J Intell Robot Syst (2009) 54:229–244
DOI 10.1007/s10846-008-9263-0

Fuzzy Logic Based Approach to Design of Flight Control and Navigation Tasks for Autonomous Unmanned Aerial Vehicles

Sefer Kurnaz · Omer Cetin · Okyay Kaynak

Received: 15 March 2008 / Accepted: 30 June 2008 / Published online: 1 October 2008

Abstract This paper proposes a fuzzy logic based autonomous navigation controller for UAVs (unmanned aerial vehicles). Three fuzzy logic modules are developed under the main navigation system for the control of the altitude, the speed, and the heading, through which the global position (latitude–longitude) of the air vehicle is controlled. A SID (Standard Instrument Departure) and TACAN (Tactical Air Navigation) approach is used and the performance of the fuzzy based controllers is evaluated with time based diagrams under MATLAB's standard configuration and the Aerosim Aeronautical Simulation Block Set which provides a complete set of tools for rapid development of detailed six-degree-of-freedom nonlinear generic manned/unmanned aerial vehicle models. The Aerosonde UAV model is used in the simulations in order to demonstrate the performance and the potential of the controllers. Additionally, FlightGear Flight Simulator and GMS aircraft instruments are deployed in order to get visual outputs that aid the designer in the evaluation of the controllers. Despite the simple design procedure, the simulated test flights indicate the capability of the approach in achieving the desired performance.

Keywords Fuzzy logic based autonomous flight computer design · UAV's SID · TACAN visual simulation

S. Kurnaz (✉) · O. Cetin
Turkish Air Force Academy, ASTIN,
Yesilyurt, Istanbul 34807, Turkey
e-mail: skurnaz@hho.edu.tr

O. Cetin
e-mail: o.cetin@hho.edu.tr

O. Kaynak
Department of Electrical and Electronic Engineering,
Bogazici University, Bebek, 80815 Istanbul, Turkey
e-mail: o.kaynak@ieee.org

1 Introduction

In the literature, it can be seen that the research interests in control and navigation of UAVs has increased tremendously in recent years. This may be due to the fact that UAVs increasingly find their way into military and law enforcement applications (e.g., reconnaissance, remote delivery of urgent equipment/material, resource assessment, environmental monitoring, battlefield monitoring, ordnance delivery, etc.). This trend will continue in the future, as UAVs are poised to replace the human-in-the-loop during dangerous missions. Civilian applications of UAVs are also envisioned such as crop dusting, geological surveying, search and rescue operations, etc.

One of the important endeavors in UAV related research is the completion of a mission completely autonomously, i.e. to fly without human support from take off to land on. The ground station control operator plans the mission and the target destination for reconnaissance and surveillance. UAV then takes off, reaches the target destination, completes the surveillance mission, and turns back to the base and lands on autonomously. In literature, many different approaches can be seen related to the autonomous control of UAVs; some of the techniques proposed include fuzzy control [1], adaptive control [2, 3], neural networks [4, 5], genetic algorithms [7] and Lyapunov Theory [8]. In addition to the autonomous control of a single UAV, research on other UAV related areas such as formation flight [6] and flight path generation [9] are also popular.

The approach proposed in this paper is fuzzy logic based. Three fuzzy modules are designed for autonomous control, one module is used for adjusting the roll angle value to control UAV's flight heading, and the other two are used for adjusting elevator and throttle controls to obtain the desired altitude and speed values.

The performance of the proposed system is evaluated by simulating a number of test flights, using the standard configuration of MATLAB and the Aerosim Aeronautical Simulation Block Set [11], which provides a complete set of tools for rapid development of detailed six-degree-of-freedom nonlinear generic manned/unmanned aerial vehicle models. As a test air vehicle a model which is called Aerosonde UAV [10] is utilized (shown in Fig. 1). Basic characteristics of Aerosonde UAV are shown in Table 1. The great flexibility of the Aerosonde, combined with a sophisticated command and control system, enables deployment and command from virtually any

Fig. 1 Aerosonde UAV

Table 1 UAV specifications

UAV specifications	
Weight	27–30 lb
Wing span	10 ft
Engine	24 cc, 1.2 kw
Flight	Fully autonomous/base command
Speed	18–32 m/s
Range	> 1,800 mi
Altitude range	Up to 20,000 ft
Payload	Maximum 5 lb with full fuel

location. GMS aircraft instruments are deployed in order to get visual outputs that aid the designer in the evaluation of the controllers.

The paper is organized as follows. Section 2 starts with the basic flight pattern definition for a UAV and then explain a sample mission plan which includes SID (Standard Instrument Departure) and TACAN (Tactical Air Navigation) procedures. A basic introduction to fuzzy control is given and the design of the navigation system with fuzzy controllers used for the autonomous control of the UAV is explained in Section 3. The simulation studies performed are explained and some typical results are presented in Section 4, and finally the concluding remarks and some plans about future work are given in Section 5.

2 UAV Flight Pattern Definition

A reconnaissance flight to be accomplished by a UAV basically contains the following phases; ground roll, lift off, initial climb, low altitude flight, climb, cruise, loiter over target zone, descent, initial and final approach and finally landing, as shown in Fig. 2. The basic flight maneuvers of the UAV during these phases are climb, descent, level flight and turns.

In this study, UAV is considered to take off and land on manually. Autonomous navigation starts when UAV reaches 2 km away form ground control station and climbs 100 m (MSL). It reaches way points in order and finishes when UAV reaches the ninth waypoint. The ninth point is the midpoint of the airfield where the landing is going to be. The definition of each point includes speed, altitude and position (longitude and longitude coordinates) values. The dashed line in Fig. 2 represents

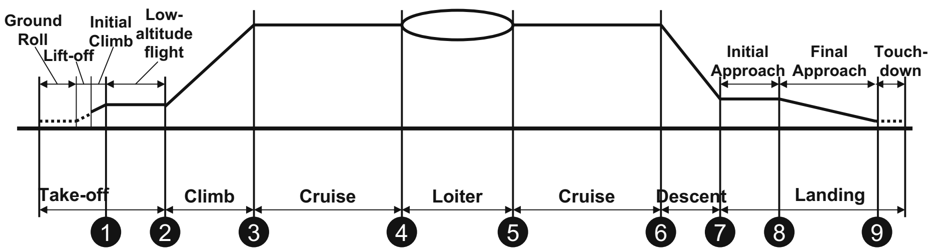

Fig. 2 Basic UAV's reconnaissance flight parts

manual control; the continuous line represents autonomous control. Character A shows the target zone location.

In this study, the test pattern includes an airport departure (Istanbul Ataturk Airport (LTBA) Turkey Runway 36L SID departure) and TACAN approach (to Yalova Airport (LTBP) runway 08) to show that UAV can fly autonomously a pattern which is designed for aircrafts if it has enough performance parameters. In classical SID and TACAN maps, a waypoint is defined with the radial angle and the distance between the VOR (VHF Omni-directional Radio Range) station and the waypoint. After transformation of the waypoints as GPS coordinates, UAV can apply SID departure and TACAN approach as a mission plan without a VOR receiver (Fig. 3).

It is presumed in this study that UAV takes off from Runway 36 and land on Runway 08. In the test pattern followed, there is a simulated target zone which is defined with three GPS coordinates. UAV visits these three points in order. While UAV is applying the flight plan, it records video when it is over the simulated target zone. After completing mission over the simulated target zone, it will try to reach IAF (Initial Approach Fix) to perform TACAN descending. UAV must obey the altitude commands for each waypoint in the plan because there are some minimal descending altitudes to avoid the ground obstacles.

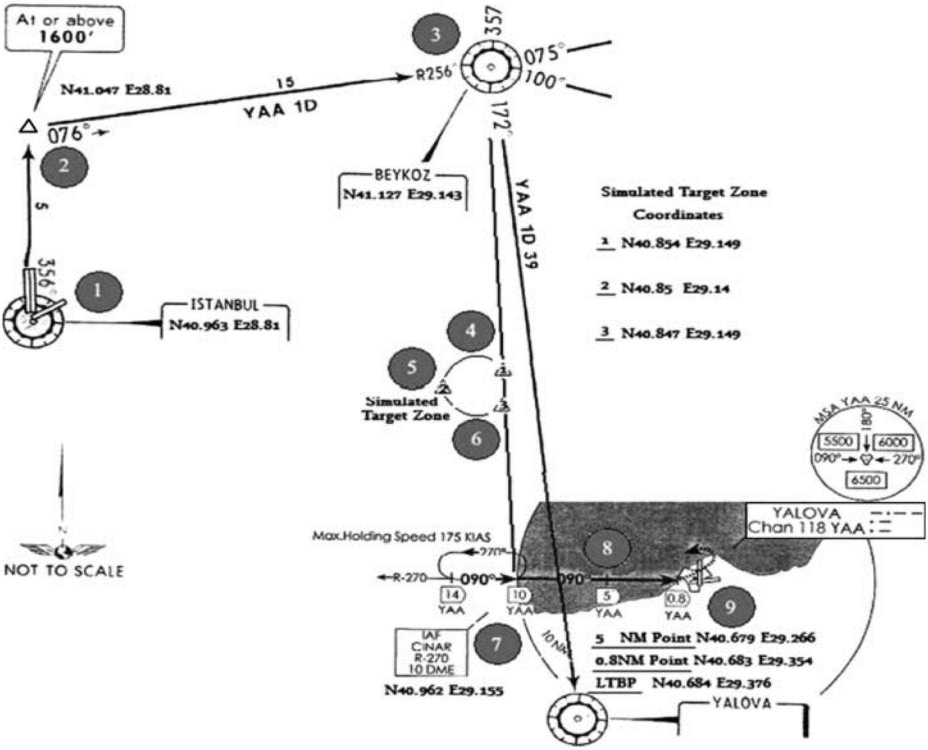

Fig. 3 UAV test flight pattern

For the test flight pattern shown in Fig. 3; the first point of the path is Istanbul Ataturk Airport which is the departure airport. After take off, the flight continues 5 NM at 356° heading while climbing to arrive at the second fix at a minimum of 1,600 ft altitude. After reaching the second fix, it turns 076° heading and fly 15 NM at YAA 1D way. When UAV reaches the BEYKOZ fix it heads south to reach the first fix of the simulated target zone. Then UAV makes a nearly half round through the three fixes of the target zone. After the completion of three target zone points, UAV turns to the seventh fix which is IAF (Initial Approach Fix) for TACAN approach for YALOVA Airport, 10 NM to the airport at MSA (Minimum Safe Altitude) of 4,000 feet. After reaching IAF, UAV turns 090° heading while descending. The eight fix is 5 NM to airport and MSA 2000'. The last fix (the ninth one) is 0.8 NM to airport; the last point for autopilot navigation. After this point if UAV operator sees the aircraft switches to the manual control, if not UAV makes a circle over airport.

3 Navigation Computer Design

As shown in Fig. 4, there are two computers in an autonomous UAV. One of them is the flight computer and the other is the mission (navigation) computer. UAV flight computer basically sets the control surfaces to the desired positions by managing the servo controllers in the defined flight envelope supplied by the UAV navigation computer as a command, reads sensors and communicates with the mission computer and also checks the other systems in the UAV (engine systems, cooling systems, etc.). In fact the navigation computer is a part of the mission computer. Because there are many duties beside navigation, like payload control, communication with GCS, etc., the navigation computer is used for flying over a pattern which is designed before the

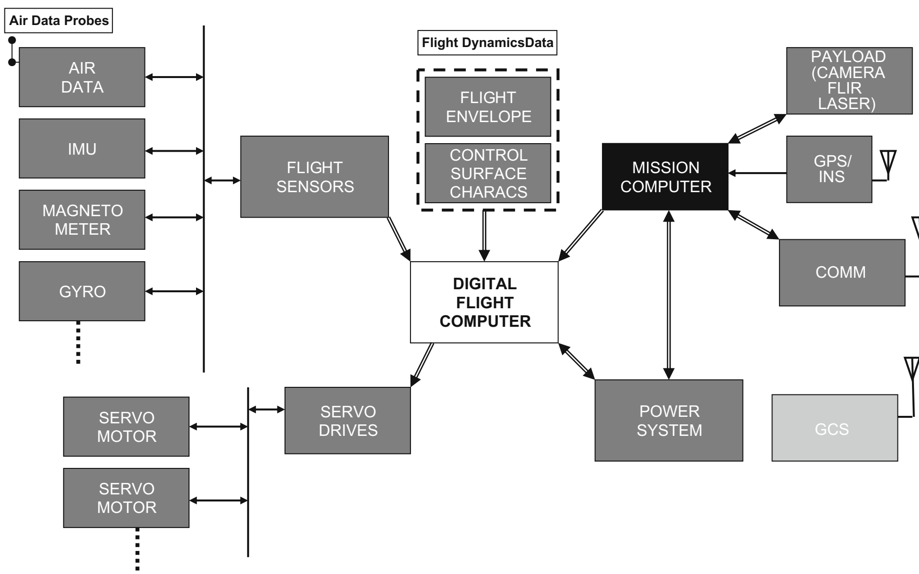

Fig. 4 UAVs electronics architecture

flight or while flying in LOS (Line Of Sight). The flight and the mission computers are always active but the navigation computer is used especially in autonomous flight. When GCS (Ground Control Station) gets control of UAV, the navigation computer goes into a passive state. During autonomous flight, the navigation computer gets the position values from sensors (GPS receiver, altimeter, INS (Internal navigation System), etc.) and then matches these values (the current position) with the desired position values (the waypoint values). The navigation computer then determines the required maneuvers of the UAV to reach the goal position and sends these data to the flight computer to apply them to control surfaces (Fig. 5).

The operation of the navigation computer proposed in this paper is fuzzy logic based. This is the main difference of the reported study with the other works seen in the literature.

Basically, a fuzzy logic system consists of three main parts: the fuzzifier, the fuzzy inference engine and the defuzzifier. The fuzzifier maps a crisp input into some fuzzy sets. The fuzzy inference engine uses fuzzy IF–THEN rules from a rule base to reason for the fuzzy output. The output in fuzzy terms is converted back to a crisp value by the defuzzifier.

In this paper, Mamdani-type fuzzy rules are used to synthesize the fuzzy logic controllers, which adopt the following fuzzy IF–THEN rules:

$$R^l\text{:If}\left(x_1 \text{ is } X_1^l\right) \text{ AND}\ldots\text{AND } \left(x_n \text{ is } X_n^l\right) \text{ THEN } y_1 \text{ is } Y_1^l,\ldots,y_k \text{ is } Y_k^l \tag{1}$$

where R^l is the lth rule $x = \left(x_{l_1} \ldots, x_n\right)^T \in U$ and $y = (y_l \ldots, y_k)^T \in V$ are the input and output state linguistic variables of the controller respectively, U, $V \subset \Re^n$ are the universe of discourse of the input and output variables respectively, $\left(X_{l_1} \ldots, X_n\right)^T \subset U$ and $(Y_1 \ldots, Y_k)^T \subset V$ are the labels in linguistic terms of input and output fuzzy sets, and n and k are the numbers of input and output states respectively.

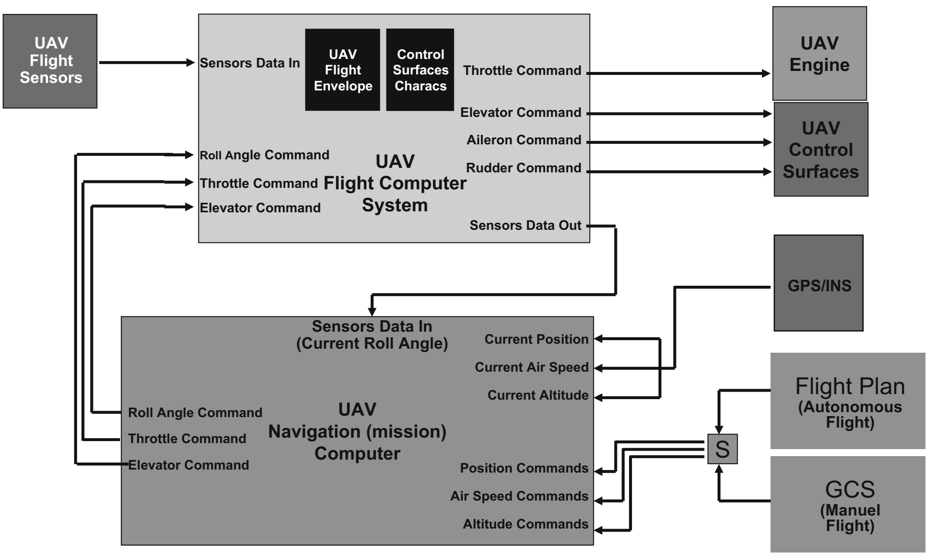

Fig. 5 Communication between UAVs Flight and Navigation Computers

We consider a multi-input and single-output (MISO) fuzzy logic controller ($k = 1$), which has singleton fuzzifier. Using triangular membership function, algebraic product for logical AND operation, product–sum inference and Centroid defuzzification method, the output of the fuzzy controller has the following form:

$$y_j = \frac{\sum_{l=1}^{M} \left(\prod_{i=1}^{N} \mu x_i^l (x_i) \right) y_j}{\sum_{l=1}^{M} \prod_{i=1}^{N} \mu x_i^l (x_i)} \tag{2}$$

where N and M represent the number of input variables and total number of rules respectively. μx_i^l denote the membership function of the lth input fuzzy set for the ith input variable.

Three fuzzy logic controllers are designed for the navigation computer in order to control the heading, the altitude and the air-speed. These three controllers acting in combination enable the navigation of the aerial vehicle (Fig. 6).

The navigation computer of UAV has four subsystems, namely heading, speed, altitude and routing subsystems. The routing subsystem calculates which way point is the next. The way point definition includes the position (the longitude and the latitude coordinates in GPS format), the speed and the altitude information of the point. When the UAV reaches the waypoint position ±01°, it means that UAV has checked that target waypoint and passes on to the next one. If the UAV cannot come within ±01° of the position of the target waypoint, it will make a circle in pattern and retry to reach that point. The routing subsystem applies this check procedure and supplies the definitions of the next waypoint to other subsystems. The inputs to the heading subsystem are the current position of the UAV (longitude and latitude coordinates in GPS format), the current roll angle (this information is received from sensors by the flight computer) and the next waypoint position which is defined by the routing system. The duty of the heading subsystem is turning the UAV to the target waypoint. The speed subsystem maintains the air speed of the UAV at the desired value. It uses the current speed and the speed command (defined by the routing system) values as inputs and its output is the throttle command. In this study, while controlling speed, the pitching angle is not used to see the effectiveness

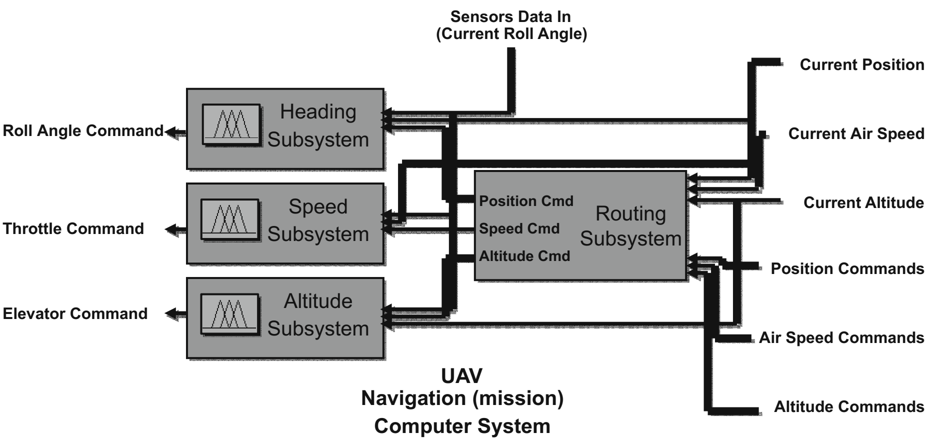

Fig. 6 Navigation computer design

of the throttle fuzzy control over UAV's air speed and to calculate the response time. The last subsystem is the altitude subsystem and it aims to maintain the altitude of the UAV at the desired value. It has the current altitude and the altitude command values as inputs and the elevator command as output.

If the fuzzy controller types in literature are reviewed, it can be seen that there are two main classes of fuzzy controllers: one is position-type fuzzy controller which generates control input (u) from error (e) and error rate (Δe), and the other is velocity-type fuzzy logic controller which generates incremental control input (Δu) from error and error rate. The former is called PD Fuzzy Logic Controller and the latter is called PI Fuzzy Logic Controller according to the characteristics of information that they process. Figure 7a and b show the general structure of these controllers.

PI Fuzzy Logic Controller system has two inputs, the error $e(t)$ and change of error $\Delta e(t)$, which are defined by

$$e(t) = y_{\mathrm{ref}} - y \tag{3}$$

$$\Delta e(t) = e(t) - e(t-1) \tag{4}$$

Where y_{ref} and y denote the applied set point input and plant output, respectively. The output of the Fuzzy Logic Controller is the incremental change in the control signal $\Delta u(t)$. Then, the control signal is obtained by

$$u(t) = u(t-1) + \Delta u(t) \tag{5}$$

As stated earlier, there are three fuzzy logic controllers in the heading, the speed and the altitude subsystems. The heading subsystem has the roll angle fuzzy logic controller, the speed subsystem has the throttle fuzzy logic controller and the altitude subsystem has the elevator fuzzy logic controller.

The throttle fuzzy logic controller has two inputs: the speed error (i.e. the difference between the desired speed and the actual speed) and its rate of change. The

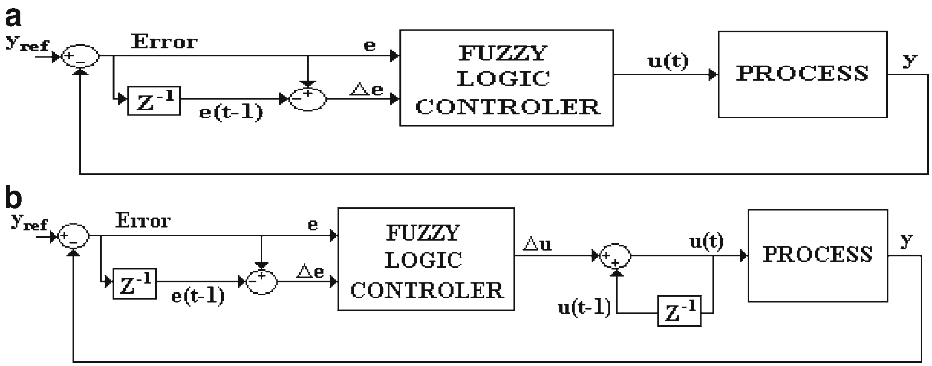

Fig. 7 **a** PD type fuzzy logic controller. **b** PI type fuzzy logic controller

Fig. 8 PID type fuzzy logic controller

latter indicates whether the UAV is approaching to the desired speed or diverging away. Like the throttle controller, the elevator control has two inputs, the altitude error and its derivative. The control output of the block is the elevator, responsible for the head going up or down. The elevator and the throttle fuzzy logic controllers are similar to the PI type fuzzy logic controller shown in Fig. 7. Because of the highly nonlinear nature of the UAV model and the inference between the controlled parameters, it is easier to aim for the required change in the control input rather than its exact value. This is the main reason for the choice of PI type fuzzy controllers.

The PI controller (also PI FLC) is known to give poor performance in transient response due to the internal integration operation. Wit a PD controller (also PD FLC) it is not possible to remove out the steady state error in many cases. Then, when the required goal cannot be reached by using only a PD or PI type of controller, it is better to combine them and construct a PID Fuzzy Controller. This is the case for roll angle control (which is used for the control of heading), and then PID type fuzzy controller was utilized for the control of heading in this work. Most commonly used PID Fuzzy Controller schematic can be seen in Fig. 8. More detailed information about PID Fuzzy Controllers can be found in [13].

While developing the fuzzy logic controllers, triangular membership functions are used for each input of the fuzzy logic controllers and simple rule tables are defined by taking into account the specialist knowledge and the experience. The control output surfaces shown in Fig. 9a, b and c are the typical ones.

As output membership functions, the throttle control output was represented with seven membership functions equally spaced in the range of [–0.02, 0.02] (frac). The membership functions used for the elevator control were rather special however, as shown in Fig. 10.

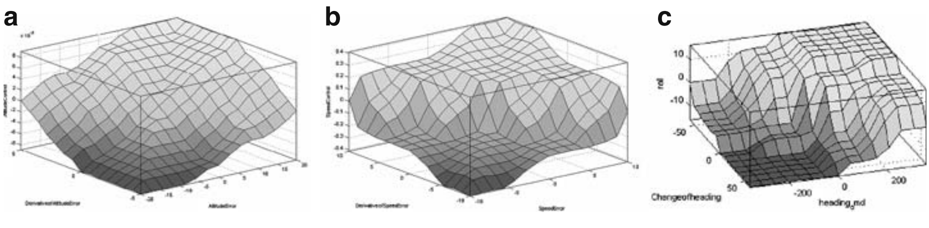

Fig. 9 **a** Rule surface of altitude. **b** Rule surface of air speed. **c** Rule surface of heading

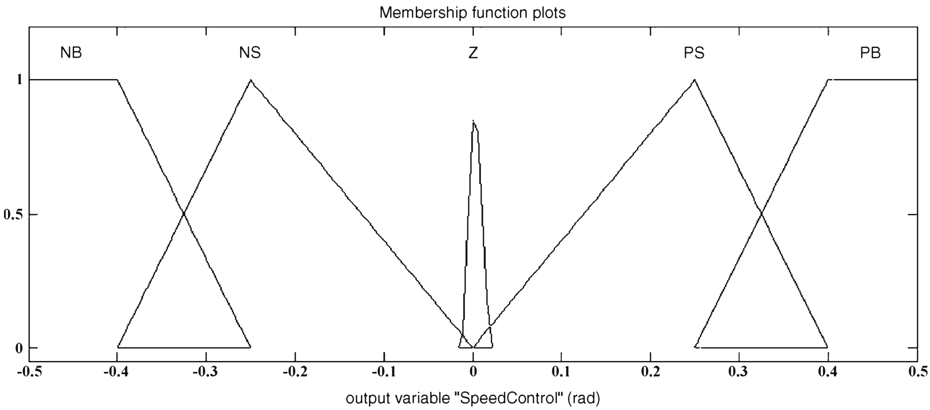

Fig. 10 Elevator control output

4 Simulation and Simulation Results

The performance and the potential of the control approach proposed are evaluated by using MATLAB's standard configuration and the Aerosim Aeronautical Simulation Block Set, the aircraft simulated being Aerosonde UAV. Additionally FlightGear Flight Simulator [12] is deployed in order to get visual outputs that aid the designer in the evaluation of the controllers (Fig. 14). Despite the simple design procedure, the simulated test flights indicate the capability of the approach in achieving the desired performance. The Simulink models that are used during the simulation studies is depicted in Fig. 11.

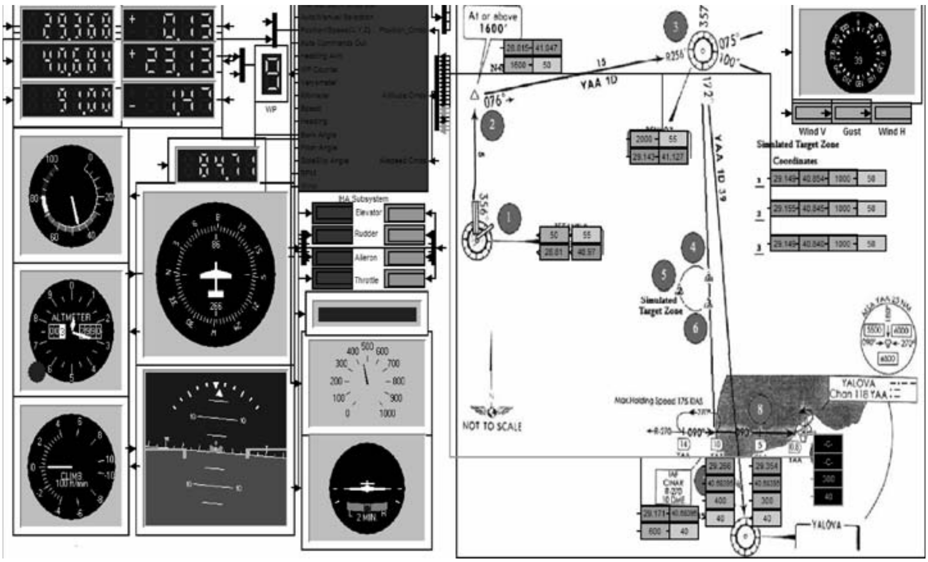

Fig. 11 Matlab Simulink simulation GMS instrument view

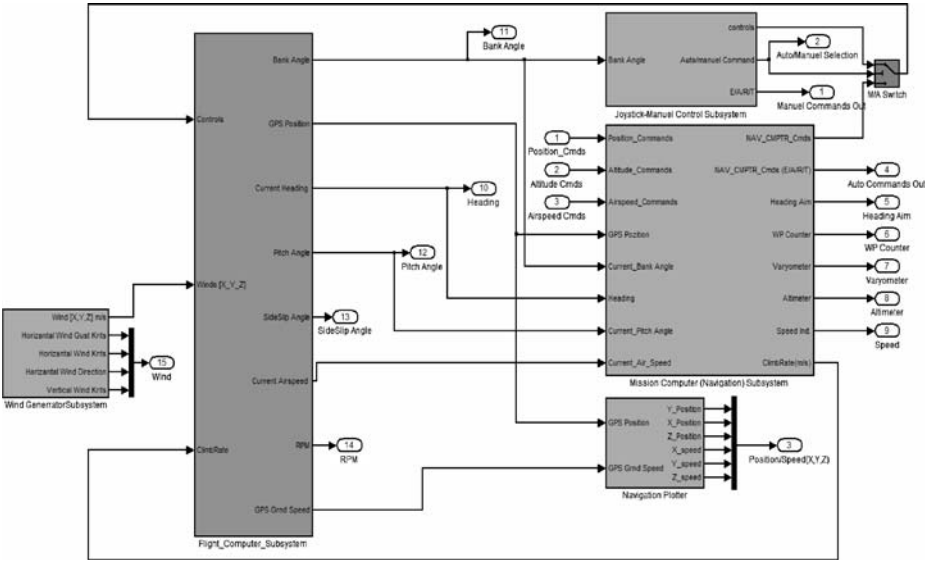

Fig. 12 Flight computer and navigation computer in simulation

In Fig. 11, the test flight pattern which includes SID and TACAN procedures is an input for the simulation. This is a kind of mission for Aerosonde UAV which includes; take of from LTBA runway 36 L and then perform SID to reach the related fixes, take video over the simulated target zone and then reach IAF to apply TACAN approach to LTBP and land on. Aerosonde UAV's current attributes can be traced over GMS aircraft instruments. These instruments are like typical aircraft instruments. A ground station control operator can manually fly the UAV by using these instruments.

As shown in Fig. 12, there are some extra modules used in simulation studies in addition to the flight computer and the mission computer modules. One of them is Joystick and Manuel control subsystem. This subsystem is used for controlling UAV by a joystick in manual selection. If the controller presses and holds the first button of the joystick, it means that the UAV is in the manual mode. When he leaves the

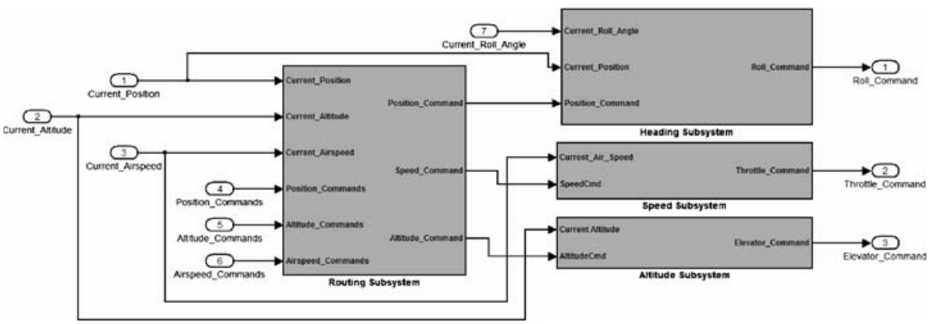

Fig. 13 Navigation computer in Simulink

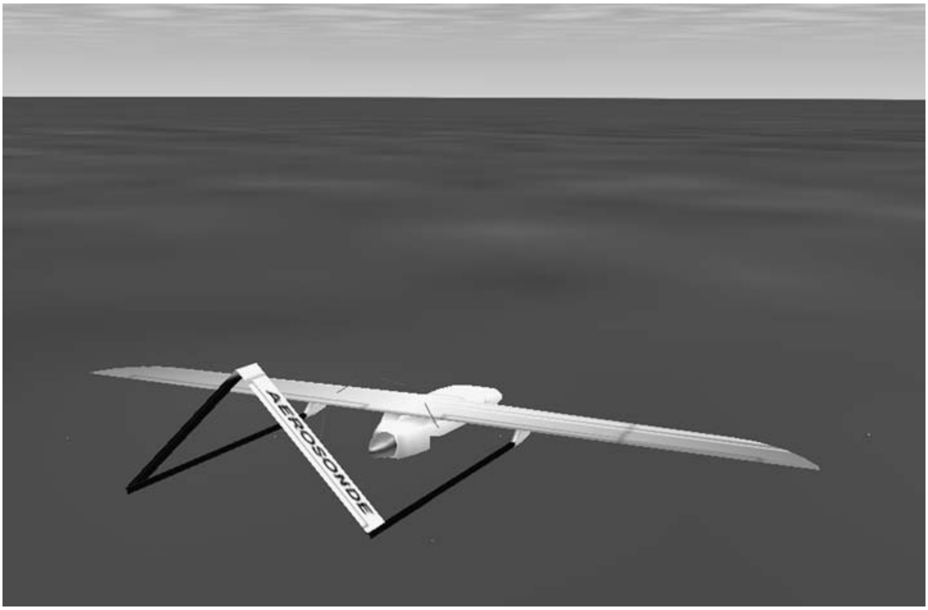

Fig. 14 Aerosonde UAV view in FlightGear while simulation is running

button, UAV assumes autonomous flight position. If the operator takes the UAV into an undesired state in manual control; after switching to autonomous control, the mission computer brings UAV to a stable position first of all and then aims to reach next waypoint. A stable position means to set the UAV's control surfaces to bring the aircraft within its limitations.

There are some limitations while controlling the UAV. These limitations are like the maximum climb rate (600 m/min), the maximum descent rate (800 m/min), the maximum speed (60 kn/h), the maximum angle of climb (25°), the maximum angle of descent (25°), the maximum roll angle (30°), etc..

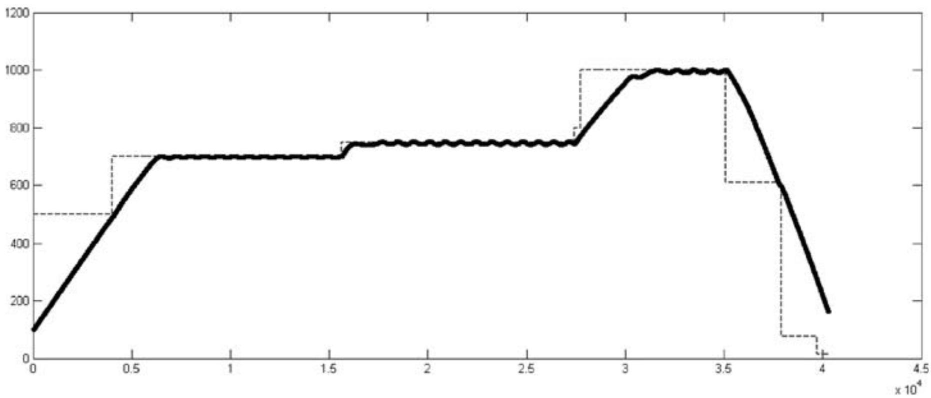

Fig. 15 Altitude–time diagram (meter/simulation time)

Another extra module used in simulation studies is the wind subsystem. The subsystem generates random wind effects as a canker to simulate the environment of UAV. The wind is represented by three dimensional vectors $[x,y,z]$. The limitations imposed in this subsystem are maximum wind speed 8 kn/h; maximum wind gust is ±2 kn/h for all altitudes. The use of the wind module enables us to see the effectiveness of the mission computer under simulated wind effects.

The last extra module in used is the navigation plotter subsystem. This module is used for generating diagrams synchronously while simulation is running to see the performance of the other modules especially that of the mission computer. These diagrams are the altitude–time diagram (Fig. 15), the heading–time diagram (Fig. 16), the speed–time diagram (Fig. 17) and the UAV position diagram (Fig. 18). By using these diagrams, one can evaluate the performance of the fuzzy controllers.

The Simulink diagram of the navigation computer is shown in Fig. 13. There are four subsystems under the navigation computer block namely routing, heading, speed and altitude subsystems. These subsystems are designed as described in Section 3 of this paper (Fig. 14).

Figure 15 depicts the change of altitude with time during test flight pattern, together with the desired altitude. It can be seen that despite the random wind effects, UAV can reach the desired altitude within range (±50 m). While doing this, the maximum angle of attack is 25° and the minimum one is 20° so that stall and possible over speeding conditions are avoided. It can therefore be stated that the fuzzy controller is successful in holding the altitude at desired levels using elevator control.

Figure 16 shows the heading response under random canker wind effects together with the desired heading command for the test flight pattern. The response time is limited because of the limitations imposed on the roll and the yaw angles. There again

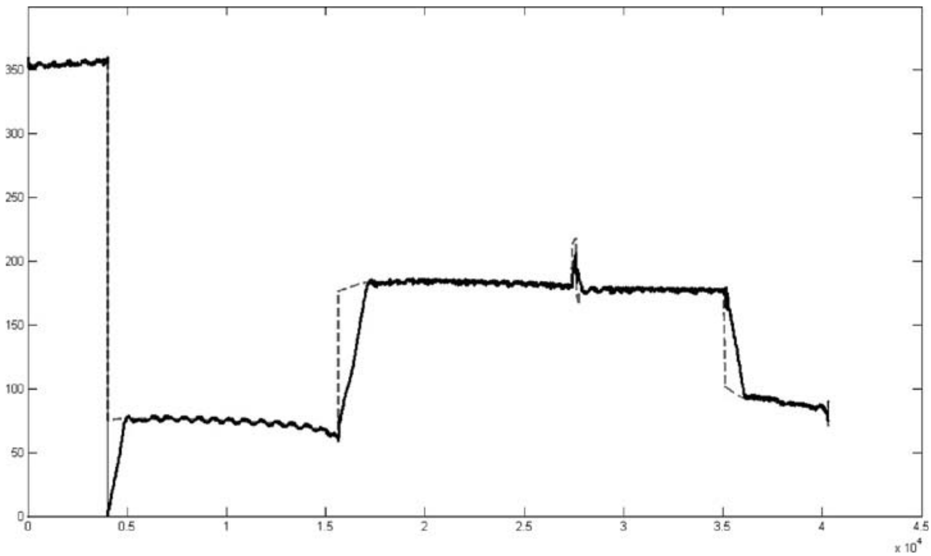

Fig. 16 Heading–time diagram (degree/simulation time)

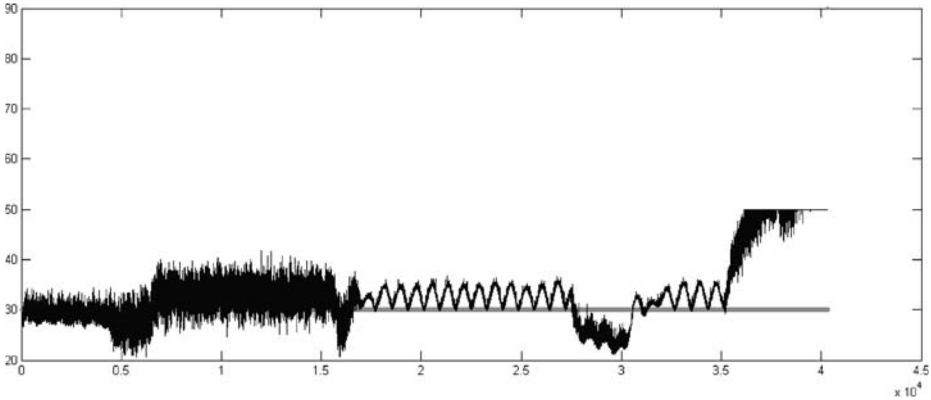

Fig. 17 Speed–time diagram (meter/second/simulation time)

the fuzzy logic controller is successful in holding the desired heading by controlling the roll angle.

The change in the speed of UAV over the test flight pattern under random canker wind effects is shown in Fig. 17, the desired speed being 30 m/s over the whole duration of the flight. The mission computer controls the speed only by throttle control not with the angle of attack. This is purposefully done to see how the throttle effects the speed and how a fuzzy logic controller controls throttle. So large errors can be seen while applying climb and descent maneuvers.

While the simulation is running a trajectory of the vehicle is generated and plotted on the navigation screen. By this way one can observe and monitor the aircraft position over a 100 km $\times$ 100 km map or a GPS coordinate based map (Fig. 18). It

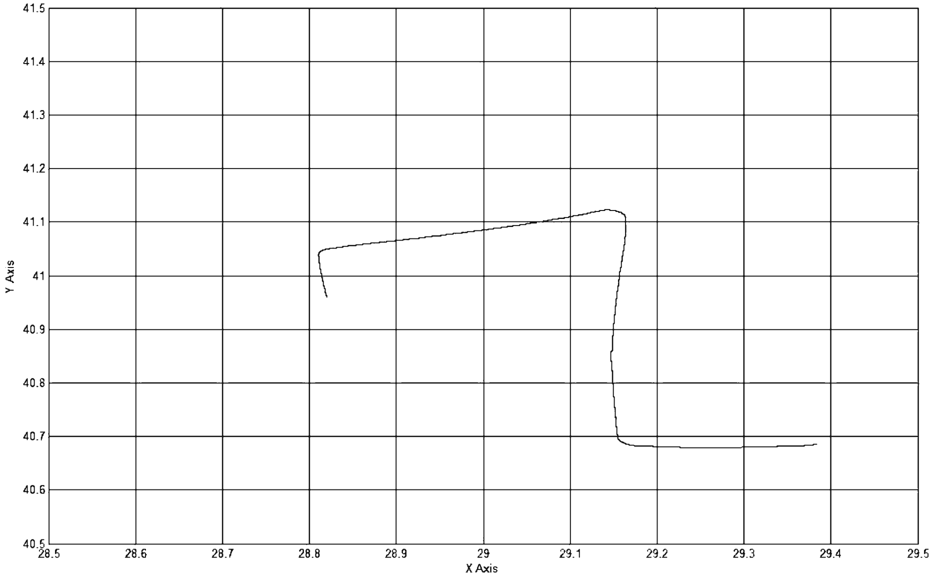

Fig. 18 UAV position diagram (latitude/longitude position)

can be seen from Fig. 18 that UAV gas completed the test flight pattern successfully under random canker wind effects. That is to say UAV reached every fix point which is defined in test flight pattern in 01° (00′ 00″ 010 as a GPS definition) error range.

5 Conclusion

The main purpose of the navigation computer is to enable the UAVs to accomplish their mission autonomously, without any (or with minimal) input from the operator. Mission computer design in this paper provides autonomy to the UAV in all phases of a typical UAV's mission except take off and land on. These provide the airplane with improved dynamic stability by regulating the flight parameters within limited ranges, at the same time tracking of UAV mission plan.

Although there are many control law architectures, the classic PID control approach augmented with online gain scheduling provides the ideal mix of robustness and performance for typical aircraft dynamics. The stability and control loops can be tuned to provide the desired performance and robustness specifications by adjusting a set of autopilot parameters or gains. But this is done through linear analysis—the nonlinear aircraft model is linearized for a representative set of flight conditions that cover the operating envelope of the aircraft. The linear dynamics of the closed-loop system (aircraft + autopilot) are analyzed in terms of stability and control responses (overshoot, settling time). By using fuzzy controllers, this difficult design process is avoided; nevertheless stable control and fast reaction time over conventional autonomous UAVs can be achieved as shown in this paper. The capability to do a dynamic planning of the desirable flight pattern is also important and this is done in this paper by using the current position of the moving UAV and the stationary target positions.

There are three key attributes for performing autonomous navigation; *perception, intelligence* and *action*. Perception means the ability of the UAV to acquire knowledge about the environment and itself. If UAV completes its mission without human support after takes off it means that UAV has intelligence attribute. And the last one is action which is the ability of vehicle to travel from point A to point B. How capable is a vehicle to perform these different functions is the metric to evaluate the degree of its autonomy. In this paper intelligence and action attributes applied by using fuzzy logic controllers successfully.

This paper also demonstrates that an UAV can apply a SID and TACAN approach if it has enough performance like an aircraft without human control. A series of SID for an airport can be planned just before the mission and the SID allowed by ATC (Air traffic controller) and meteorology can be applied autonomously. UAVs can to apply a TACAN approach too if they are properly instrumented with VOR and TACAN receivers. Without a VOR receiver, a SID can be applied by using GPS. Based on these attributes, fuzzy logic has been identified as a useful tool for developing controllers for UAVs so that they will be able to perform autonomous navigation.

In this paper, a fuzzy logic based autonomous flight controller for UAVs is proposed. The simulation studies presented verify that the UAV can follow the predefined trajectories despite the simplicity of the controllers. However, as seen by the simulation results, there exist some oscillations and errors when wind effects are added to the simulation environment. In future studies, the goals are to develop algorithms for a better tuning of membership functions which utilize the well known ANFIS (Adaptive Neuro Fuzzy Inference Systems) approach or possibly type-2 fuzzy sets. Autonomous take off and land on will also be tried.

References

1. Doitsidis, L., Valavanis, K.P., Tsourveloudis, N.C., Kontitsis, M.: A framework for fuzzy logic based UAV navigation and control. In: Proceedings of the International Conference on Robotics Automation, vol. 4, pp. 4041–4046 (2004)
2. Schumacher, C.J., Kumar, R.: Adaptive control of UAVs in close-coupled formation flight. In: Proceedings of the American Control Conference, vol. 2, pp. 849–853 (2000)
3. Andrievsky, B., Fradkov, A.: Combined adaptive autopilot for an UAV flight control. In: Proceedings of the 2002 International Conference on Control Applications, vol. 1, pp. 290–291 (2002)
4. Dufrene, W.R., Jr.: Application of artificial intelligence techniques in uninhabited aerial vehicle flight. In: The 22nd Digital Avionics Systems Conference vol. 2, pp. 8.C.3–8.1-6 (2003)
5. Li, Y., Sundararajan, N., Sratchandran, P.: Neuro-controller design for nonlinear fighter aircraft maneuver using fully tuned RBF networks. Automatica **37**, 1293–1301 (2001)
6. Borrelli, F., Keviczky, T., Balas, G.J.: Collision-free UAV formation flight using decentralized optimization and invariant sets. In: 43rd IEEE Conference on Decision and Control vol. 1, pp. 1099–1104 (2004)
7. Marin, J.A., Radtke, R., Innis, D., Barr, D.R., Schultz, A.C.: Using a genetic algorithm to develop rules to guide unmanned aerial vehicles. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, vol. 1, pp. 1055–1060. (1999)
8. Ren, W., Beard, R.W.: CLF-based tracking control for UAV kinematic models with saturation constraints. In: Proceedings of the 42nd IEEE Conference on Decision and Control, vol. 4, pp. 3924–3929 (2003)
9. Dathbun, D., Kragelund, S., Pongpunwattana, A., Capozzi, B.: An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments. In: Proceedings of the 21st Digital Avionics Systems Conference vol. 2, pp. 8D2-1–8D2-12 (2002)
10. Global Robotic Observation System, Definition Of Aerosonde UAV Specifications: http://www.aerosonde.com/aircraft/
11. Unmanned Dynamics, Aerosim Aeronautical Simulation Block Set Version 1.2 User's Guide: http://www.u-dynamics.com/aerosim/default.htm
12. FlightGear Open-source Flight Simulator. www.flightgear.org
13. Qiau, W., Muzimoto, M.: PID type fuzzy controller and parameter adaptive method. Fuzzy Sets Syst. **78**, 23–35 (1995)