Proceedings of the IEEE
International Conference on Mechatronics & Automation
Niagara Falls, Canada • July 2005

A Flight Control and Navigation System of a Small Size Unmanned Airship

Jinjun Rao, Zhenbang Gong, Jun Luo and Shaorong Xie

Department of Precision Mechanical Engineering
Shanghai University
P. O. Box 108, Yanchang Rd. 149, Shanghai 200072, P. R. China
raojinjun@163.com; zhbgong@mail.shu.edu.cn; wqlj1228@263.net; srxie@263.net

Abstract **– Unmanned airships present an enormous potential for low-speed and low-altitude exploration applications. In order to develop a flight robot based on a small size unmanned remotely controlled airship, a flight and navigation system is presented in this paper. The hardware and software of the system are designed. As the kernel of the system, the flight control and navigation strategies are detailed. On the basis of the airship dynamic model, the architecture of the control system is presented, and the fuzzy controllers are employed and optimized using an improved genetic algorithm. Based on the airship model and the control and navigation strategies, the simulation studies of 3D path tracking problem are given and analyzed.**

Index Terms – Airship, Fuzzy control, Navigation, Genetic algorithm, Unmanned aerial vehicle

I. INTRODUCTION

Airships, also known as lighter-than-air (LTA) aerial vehicles, outperform fixed-wing vehicles (airplanes) and rotary-wing aircrafts (helicopters) in stability in low speed, operation safety, long endurance, payload to weight ratio, etc. [1-2]. Robotic airships are the LTA Unmanned aerial vehicles (UAVs) with high levels of autonomy, and they have enormous application potential, including advertising, VIP security inspection, traffic monitoring and management and so on. As a result, increasing researcher prefer to taking them as platforms to study information collecting, environmental monitoring, intelligent control and navigation [3-7].

We are engaged in studying a small size unmanned airship at present. Our goal is to develop a robotic airship with significant levels of autonomy, especially in cruise and mission phases of its operation. For this purpose, the flight control and navigation system is designed, which focus primarily on sensing, control and navigation. We believe that this system provides an important contribution to the study of semi-autonomous robotic airships.

II. ARCHITECTURE OF THE SYSTEM

This non-rigid airship has a length of 11.8m, a maximum diameter of 3m and a volume of 50 m^3. It is equipped with two engines on both sides of the gondola, and has four control surfaces at the stern, arranged in '+' configuration. Its payload capacity is around 15kg at sea level, and can fly with maximum speed of about 60 km/h, remotely piloted by operator without autonomous flight control and navigation system.

To transform the airship from a radio-controlled machine to a semi-autonomous flight robot, the flight control and navigation system hardware and software are developed.

A. *Hardware*

The hardware includes ground part and onboard part. The ground part is composed of ground host (generally laptop PC) and RC transmitter. The data transmission units (DTUs) are utilized to communicate between onboard part and ground part. A miniature color camera with high resolution is used to gather the video information, and the information is transmitted via video link and displayed in the monitor on the ground. The onboard part is the kernel of the system, consisting of a set of sensors, actuators and MCU as follows:

1) Airship state observer: In order to allow automatic flight control, the following sensors are used: a GPS receiver, a digital compass that provides the airship heading (yaw), pitch and roll angles, two piezoelectric vibrating gyros that provide the pitch and yaw rates. Besides, an altimeter and a speedometer, both based on silicon piezoresistive pressure sensors, are used for helpful environment information. For safety consideration, the battery status is detected, too.

2) Actuator: The control surfaces servos of the airship are usual PWM controlled devices. Servos' tillers rotate to specific angle based on the received PWM signals. The elevator, rudder and throttle channels can be controlled by the system. For safety reasons, and to enable a mixed manual/automatic control, an analog switch is used to select between MCU and RC control for each actuator.

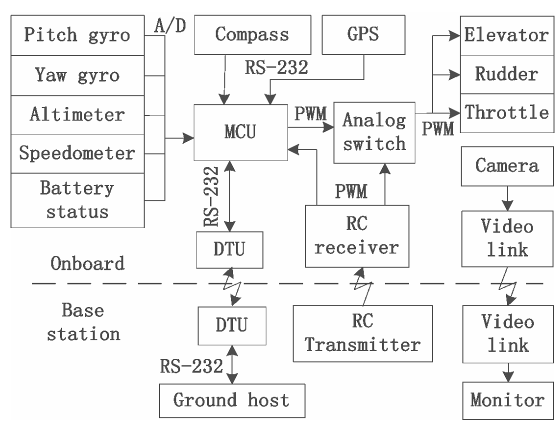


Fig. 1 The hardware structure of the flight control and navigation system of the small size unmanned airship

0-7803-9044-X/05/$20.00 © 2005 IEEE

3) *MCU*: Aforementioned sensors, actuators and digital communication device have different signals and interfaces. And, high real-time performance is required in the flight control system. Additionally, the low cost, low power waste and small volume are developing trends of the flight control system. Consequently, it's important to choose the fittest processor in the onboard system. We opted for a mixed-signal System-on-a-Chip MCU C8051F12x, featured in high speed (up to 100 MHz) and integrated with all the necessary communication ports.

The whole hardware architecture is sketched in Fig. 1, where the various formats of information exchanges between the components are shown, also.

B. Onboard Software

The software of the onboard system is complicated due to the complexity of the autonomous airship. In this system, all codes are implemented in C, and are compiled, linked and stored in the in-system programmable FLASH memory of the MCU. Its functions include:

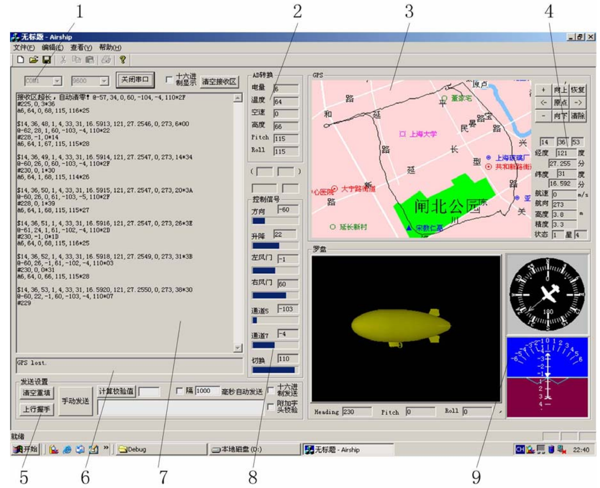
1) Signals Processing: The sensors and effectors interface with MCU in different ways, and some of these devices require high real-time performance. Since the MCU has high speed and support 20 interrupt source, all signals of sensors, RC receiver, radio link and actuators are processed in the corresponding interrupt service routine (ISR). In this soft way, high real-time performance is achieved, and sensors data with frequency of up to 20 Hz can be collected. It's enough for airships because of their relatively large time constants. In order to eliminate the disturbing noise and get the right observation, many digital filtering algorithms have been tested and employed, also.

2) *Fault Tolerance and Fault Diagnosis*: To enhance the robustness of the software, the capabilities of fault tolerance and fault diagnosis are integrated. The software can handle possible faults, such as: GPS position fix is invalid in shielded or strong electro-magnetic interference circumstance; compass outputs invalid data in abnormal magnetic field; the batteries power is low; etc. Once fault is detected, the software will respond in presetting way. For an instance, when a wrong ground command is received, this command will be omitted, and a relevant prompt will be transmitted to remind the user.

3) *Execution of Ground Command*: Generally, the ground users communicate with the airship for following three purposes: a) to read the flight data recorded during last flight test; b) to change control parameters and test different control strategies; and c) to change waypoints. The flight data can be read from nonvolatile SRAM and transmitted to the ground station if a relevant request is occurred in 10 seconds after startup of onboard system. The other two functionalities can be done even in flight. They are very important for the safe development and test of new low-level control, navigation, and intelligent mission control strategies.

4) *Flight Control and Navigation*: The flight control and navigation strategies are developed and detailed in Section III.

Besides, communication protocol, flight mode switch, etc, are also designed and integrated.



1) COM setting; 2) A/D data; 3) Digital map and flight trajectory; 4) GPS data; 5) Command editor; 6) Error prompt; 7) Flight data; 8) control inputs; 9) Flight attitude

Fig. 2 The HMI of the monitoring system

C. Human-Machine Interface

The human-machine interface (HMI) of ground station provides the communication and visualization mechanism between the operator and the airship [8]. The operator uses this interface to receive and visualize flight data acquired onboard, and to define the control parameters and navigation profile. This interface is developed using Visual C++ 6.0 object-oriented programming environment, and its main functions are detailed in the following.

1) Flight data display and storage: The flight data from onboard system is displayed in corresponding textboxs and stored in a file appointed by user at the same time. The stored data provide the first-hand information for further flight investigation and analysis.

2) Ground command editor: It is provided to edit ground command, calculate the checksum and transmit to onboard system. It's necessary for user to define flight profiles and change control parameters. During mission execution, if any unforeseen event occurs which can not be automatically solved by the airship, the ground operator intervene through this editor.

3) Flight path visualization: The vehicle flight path and geographic information can be shown in a longitude-by-latitude window. It is capable of zooming, panning, and freezing so that users can examine in detail the flight data.

4) Airship attitude visualization: The airship attitude is shown in two ways simultaneously. One way is to use avionics panel to present a representation of compass data, i.e. angles of heading, pith and roll of the airship. The other way is to model 3D geometrical body using OpenGL libraries to of the airship.

Fig. 2 is a view of the HMI of the system in use. Through this interface, all interested data of the robotic airship in flight can be obtained and supervised explicitly, and interaction between human and airship is also convenient.

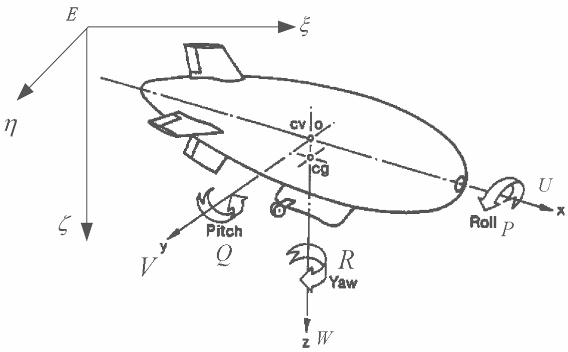


Fig. 3 Coordinate and symbols of airship

III. Flight Control and Navigation

A. Airship Modeling

As the basis of development of the control system, the 6-DOF dynamics model is necessary for studying airship control and navigation strategies at low cost. This model assumes that motion is referenced to an orthogonal body-fixed frame $R(o,x,y,z)$. Its origin is at the center of volume (cv), assumed to coincide with the gross center of buoyancy (Fig.3). The position and orientation of R with respect to Earth-fixed frame $R_0(E,\xi,\eta,\zeta)$ is obtained through (X , Y , Z) and the Euler angles (ϕ , θ , ψ). The airship linear and angular velocities are given by (U , V , W) and (P , Q , R), respectively.

The dynamic model can be stated by Equ.(1) and Equ.(2). Equ. (1) describes the system dynamics with respect to R, while Equ. (2) represents the kinematic link between the frames R and R_0:

$$M\dot{X} = F_d + F_a + G + P \tag{1}$$

$$\dot{S} = JX \tag{2}$$

where M is the 6×6 mass matrix and includes both the actual inertia of the airship and the added inertia elements associated with the dynamics of buoyant vehicles; X =[U , V , W , P , Q , R] ^{T} is the vector of airship state variables; F_d is the 6×1 dynamics vector containing the Coriolis and centrifugal terms; F_a is the 6×1 vector of aerodynamic forces and moments; G is the 6×1 gravity vector, which is a function of the difference between the weight and buoyancy forces, and P is the 6×1 vector of propulsion forces and moments; S =[X , Y , Z , ϕ , θ , ψ] ^{T} is the airship kinematic variables; J is the transition matrix from R to R_0.

The airship dynamics indicates that the state parameters involved in longitudinal and lateral motions are weakly dependent. So the system can be split into two subsystems in the following way [3]:

1) $S_{long} = [X,Z,\theta]^T$ and $X_{long} = [U,W,Q]^T$ to describe the dynamics within the longitudinal plane, the control inputs are deflection of elevator δ_e and deflection of throttle δ_t.

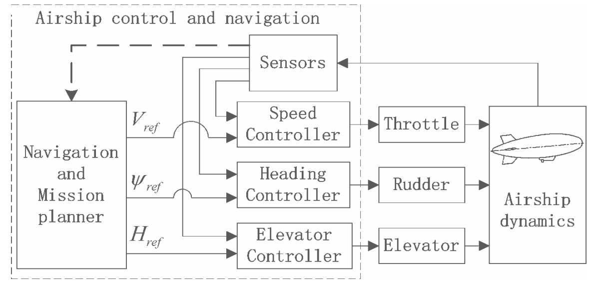


Fig. 4 The architecture of control and navigation system of the unmanned airship

2) $S_{lat} = [Y,\phi,\psi]$ and $X_{lat} = [V,P,R]^T$ to describe the dynamics within the lateral plane, the control input is deflection of rudder δ_r.

The airship dynamics model shows that: 1) the rolling corresponding mode is structurally stable and not controllable [9]; 2) the longitudinal and lateral control can be viewed as decoupled [10]; but 3) airship has more nonlinearities than ordinary aircraft due to the added mass [11].

B. Architecture of Control and Navigation System

For the airship, speed, heading, pitch and height of the vehicle are under control by the low level flight control system. According to the decoupled lateral and longitudinal dynamics model, the control architecture of the system is presented as Fig. 4.

In this architecture three independent controllers are utilized as follows: 1) A proportional-integral controller for the longitudinal velocity v acting on the throttle deflection δ_t; 2) A heading controller acting on the rudder deflection δ_r; 3) A controller for height and pitch acting on the elevator deflection δ_e. The navigation and mission planner is designed to provide longitudinal velocity reference V_{ref} , height reference H_{ref} and heading reference ψ_{ref} . In a specific mission flight, V_{ref}, H_{ref} and the waypoints are predefined by user. As the airship position is motional, so the planner should compute ψ_{ref} in real-time for heading controller.

C. Navigation

The position of airship provided by GPS module is given in world geodetic system (WGS, represented in longitude /latitude). In order to obtain ψ_{ref}, coordinate transformation should be conducted from WGS to Earth-fixed local system (LS, a right handed orthogonal system with X axis in East and Y axis in North). Assume the origin of the LS has coordinates (λ_0, ϑ_0) in WGS, the point with coordinates (λ , ϑ) in WGS has coordinate (x, y) in LS:

$$\begin{cases} x = l_E \cos\vartheta_0(\lambda - \lambda_0)/360 \\ y = l_M(\vartheta - \vartheta_0)/360 \end{cases} \tag{3}$$

where l_E is the average length of the Earth equator and l_M is the average length of the Earth meridian.

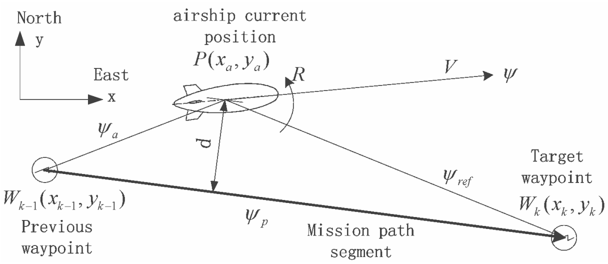


Fig. 5 Symbols of navigation

In straight line path tracking problems [12-13], the vehicles are required to flight through a set of predefined waypoints at a given constant speed and altitude. It's a good application for testing autonomy level of robotic UAVs. As depicted in Fig. 5, the airship current position, previous and current target waypoints are $P(x_a, y_a)$, $W_{k-1}(x_{k-1}, y_{k-1})$ and $W_k(x_k, y_k)$, respectively; V and R are the linear ground speed and yawing angular velocity of the airship, respectively; ψ is the heading of the vehicle; ψ_a, ψ_p and ψ_{ref} are the directions from W_{k-1} to P, from W_{k-1} to W_k and from P to W_k, respectively, and they can be computed out using the positions information through inverse trigonometric function. d is the lateral tracking error to the desired path, and its sign is the same as (ψ_a - ψ_p). In the real time system, if distance between airship and target waypoint is less than 20m, the navigation and mission planner regards that the n^{th} waypoint is arrived and the $(n+1)^{\text{th}}$ waypoint is set as the target point. If the last pre-defined way-point is arrived, the first one is turned again.

D. Heading Controller

The control block of heading controller is shown in Fig. 6. The heading controller is consists of a fuzzy controller and an integrator.

As for the integrator (Fig. 6(b)), it is used to include the integral of the error as a third input to the heading controllers to compensate setpoint error caused by disturbances. The integrator is reset to zero on each change of setpoint. Because integration only occurs for small values of error, the problems of integrator windup are avoided while still eliminating setpoint error [14].

The fuzzy controller (Fig. 6(a)) is the main part of the heading controller. Its inputs are heading error E and heading error rate EC, and the output is δ_r. K_e, K_c normalize the universes of discourse of the inputs to the range of [-1, 1]. The universe of discourse of output deflection is limited in ±30 degree by the actual mechanism of the control surfaces, so K_d =30. Seven fuzzy sets are defined for each input variable, as shown in Fig. 7, where x_1=0.1 and x_i=0.3 (i=2, 3, …, 7) for the initial design. And the rule base is built as Table I.

E. Elevator Controller

The elevator controller consists of two controllers in a two-layer structure (Fig. 8). The first layer is height controller, whose output is the pitch reference θ_{ref}, also the next layer pitch controller's input. The pitch controller output elevator deflection δ_e. In another words, the height and pitch of airship are governed by elevator. Both height controller and pitch controller have the same control block as the heading controller.

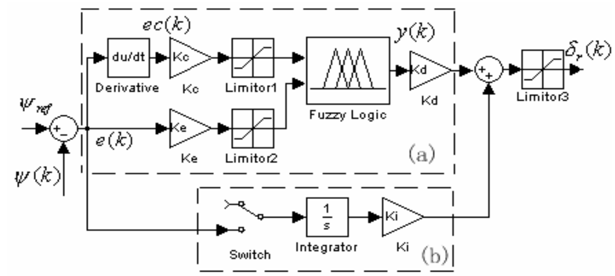


Fig. 6 Heading controller block diagram

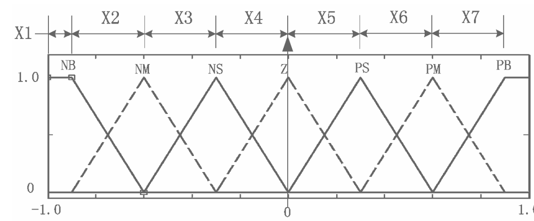


Fig. 7 The membership functions for the fuzzy input variables

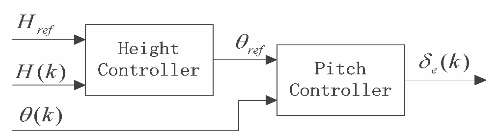


Fig. 8 Elevator controller

TABLE I
THE RULE BASE OF FUZZY CONTROLLERS

E \ EC	NB	NM	NS	Z	PS	PM	PB
NB	-0.8333	-0.8333	-0.6333	-0.5	-0.3333	-0.1667	0
NM	-0.8333	-0.6333	-0.5	-0.3333	-0.1667	0	0.1667
NS	-0.6333	-0.5	-0.3333	-0.1667	0	0.1667	0.3333
Z	-0.5	-0.3333	-0.1667	0	0.1667	-o.3333	0.5
PS	-0.3333	-0.1667	0	0.1667	0.3333	0.5	0.6333
PM	-0.1667	0	0.1667	0.3333	0.5	0.6333	0.8333
PB	0	0.1667	0.3333	0.5	0.6333	0.8333	0.8333

F. Optimization of Fuzzy Controller Using Improved GAs

There are three fuzzy controllers used in the system. Though they are robust for the plant and easy to design, the qualities of control may be not that good, since the rule base and membership functions of fuzzy set are determined by designers imprecise. So a tuning operation is needed for the fuzzy control system. In fact, this operation is a process of optimization. Genetic algorithms (GAs), known to be robust general-purpose global optimization method, are utilized to optimizing the membership functions of fuzzy controllers.

Considering Fig. 7, the membership functions of two fuzzy input variables are determined by parameters $\mathbf{x} = (x_1, x_2, ..., x_{14})$, (where, $x_1, x_2, ..., x_7$ for error and

$x_7, x_8, \ldots, x_{14}$ for error rate) of a controller. In this approach constraint conditions are inducted to guarantee that all fuzzy sets are in the universes of discourse:

$$\begin{cases} g_1 = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 - 2 \le 0 \\ g_2 = x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} - 2 \le 0 \end{cases} \quad (4)$$

In traditional GAs, the optimization problems with constraint conditions are converted into the ones without constraint conditions using penalty functions. But it's not easy to determine the penalty coefficients [15]. When the penalty coefficients are small, some individuals out of the searching space may have high fitness, so the GAs may get the wrong results. While when they are too huge, the differences among individuals are weak, so it's hard for the selection operator of GAs to select valid individuals with high fitness. Obviously the traditional GAs are expected to be improved for the constraint optimization problems.

A selection operator of GAs based on direct comparison approach is presented.

Step1: A function measuring the degree of the individual $\mathbf{x}$ violating the s.t. m is defined, for example:

$$m(\mathbf{x}) = -\varepsilon + \sum g_j(\mathbf{x}) \quad (5)$$

where ε is a small positive constant.

Step2: Choose two individuals, say $\mathbf{x1}$ and $\mathbf{x}2$, from previous generation randomly.

Step3: Select one to the next generation follow two rules: if $m(\mathbf{x1})$ and $m(\mathbf{x}2)$ have the same signs, the one with smaller objective function value is selected; or if $m(\mathbf{x}1)$ and $m(\mathbf{x}2)$ have different signs, say, $m(\mathbf{x}1) < 0$, then $\mathbf{x}1$ is selected.

Repeat *Step2* and *Step3* till the next generation has enough individuals.

This operator treats constrains not by penalty functions but by directly comparison, so the advantages of GAs are preserved. Additionally, because it takes the effect of invalid solutions into consideration, the searching abilities of GAs is augmented, also.

Based on 6DOF nonlinear dynamics model of the unmanned airship, the simulation and optimization program is developed in MATLAB environment. The *ITAE* performance criterion is used as the objective function:

$$\min f(\mathbf{x}) = \int_0^T t \mid e(t) \mid dt \quad (6)$$

where e is the control error, and T is the maximal acceptable setting time of the controller. The optimum membership functions of the input variables of three fuzzy controllers are obtained. Limited by the length of the paper, only the result about heading controller is given. The optimal membership functions of heading error and heading error rate are shown in Fig. 9. Considering the stepinput of heading error, the airship responses under the optimal fuzzy heading controller and the initial controller are shown in Fig. 10. Obviously the optimal fuzzy heading controller spends much shorter time than the initial one, and overshoot is avoided.

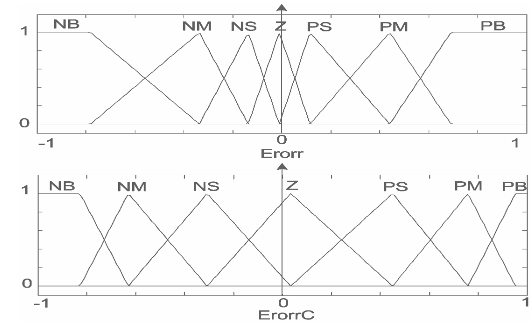


Fig. 9 The optimum membership functions of input variable of heading controller.

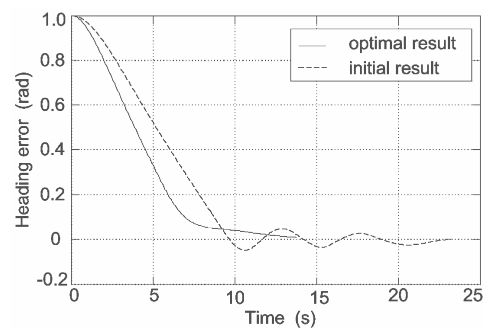


Fig. 10 Airship heading responses under the initial fuzzy heading controller and the optimum fuzzy heading controller

G. *Verification of the Control and Navigation Strategies*

In order to verify the proposed control and navigation strategies, simulations based on the dynamics model is researched. A 3D mission path, consisting of four waypoints, (0,0,0), (75,120,60), (0,240,120), (100,50,10), is chosen (the dashed line in Fig.11). This path is a challenge to the control and navigation system, because the yawing angle ranges form -64deg to 144deg, with pitch angle form -10deg to 25deg, simultaneously. The short mission paths add more difficulties, also. In this case, the trim airspeed is 8m/s. In Fig.12, because of the large time constant and large virtual mass of airship, there are tracking errors about 55m occurred in two sharp angles despite the saturate control of rudder (±30deg) is acted. Relatively, the control of elevator is slight and steady due to the small change in pitch. The results manifest that the strategies are feasible, and the system can track mission path with satisfactory precision.

IV. Conclusions and Outlook

This paper presents a flight control and navigation system for a small size unmanned airship. Firstly, the hardware and software of the system are designed. The onboard hardware consists of airship state observer, actuators, MCU, etc.. The onboard software functions include signals processing, fault tolerance and diagnosis, ground command execution, flight control and navigation. As a human-machine interface, the base station software provide a display and communication mechanism between actors and airship. Then, the control and navigation strategies are detailed. Based on the airship

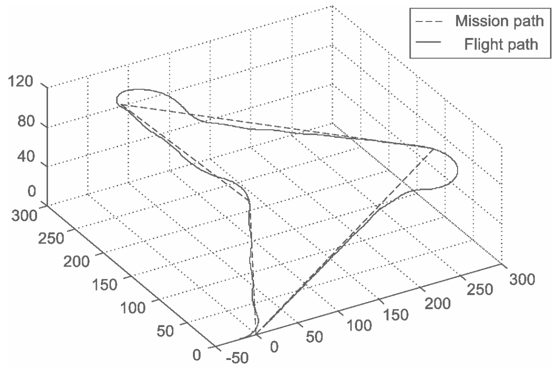


Fig. 11 The flight path and the mission path

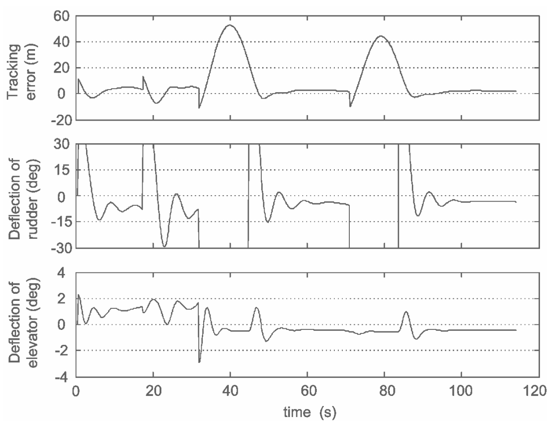


Fig. 12 Tracking error and deflections of rudder and elevator

dynamic model, the architecture of the control system is presented. Heading, height and pitch are controlled by three controllers, respectively. Each controller consists of a fuzzy controller as the main part and an integrator for compensation. For optimum performances of fuzzy controllers, the membership functions of input variables are optimized using an improved GA with a new selection operator. The simulation studies of 3D mission path tracking based on nonlinear dynamics model are given. The results manifest the control and navigation strategies are feasible.

At present, the authors are undertaking flight experiments using the presented flight control and navigation system. The tests can be carried out expediently thanks to the user-friendly HMI. And the parameters require little tune, which is mainly owed to the implementation of the optimized fuzzy strategy. We expect to develop a robotic airship with higher level of autonomy using vision information for target identification and tracking.

ACKNOWLEDGMENT

The research is financially supported by National Hightech. Research and Development program under Grant 2001AA422160 and Grant 2004AA420110, and by Shanghai Development Fund of Science and Tech. under Grant 02DI11019, and by "Mechatronics Engg." State Key Discipline Fund and Shanghai Key Discipline Fund (SUB67) of Shanghai University.

REFERENCES

[1] A. Elfes, S. S. Bueno, M. Bergerman, J.J. G. Ramos, S. B. V. Gomes, "Project AURORA: Development of an Autonomous Unmanned Remote Monitoring Robotic Airship", *J. Braz. Comp. Soc.* Vol. 4 no. 3, Apr. 1998, pp70-78

[2] A. Elfes, S. Bueno, S.S. Bueno, M. Bergerman, E. Paiva, J.J.G. Ramos and J.R. Azinheira, "Robotic airships for Exploration of Planetary Bodies with an Atmosphere&colon: Autonomy Challenges", *Autonomous Robots*, Vol.14 no.2-3, pp147-164

[3] P. Rives, J. R. Azinheira, "Linear structures following by an airship using vanishing point and horizon line in a visual servoing scheme," in *Proc. 2004 IEEE Int. Conf. on Robotics & Automation, ICRA'04*. vol. 1, pp. 255-260. April, 2004.

[4] S. Zwaan, A. Bernardion, J. Santos-Victor, "Visual station keeping for floating robots in unstructured environments," *Robotics and Autonomous Systems*, vol.39, pp. 145–155, New Orleans, LA, Jane, 2002.

[5] G. Baumbach, U. Vogt, "Experimental determination of the effect of mountain-valley breeze circulation on air pollution in the vicinity of Freiburg", *Atmospheric Environment*, Vol.33, no.24-25, Oct. 1999, pp4019-4027

[6] J.B. Ries, I. Marzolff, "Monitoring of gully erosion in the Central Ebro Basin by large-scale aerial photography taken from a remotely controlled blimp", *Catena*, Vol. 50, no. 2-4, Jan. 2003, pp309-328

[7] G. Xia and D. R. Corbett, "Cooperative Control Systems of Searching Targets Using Unmanned Blimps," *Proceeding of the 5th World Congress on Intelligent Control and Automation*, Jun. 2004, pp1179-1183.

[8] Ramos, J.J.G., Maeta S.M., Mirisola L.G.B., S. S. Bueno, M. Bergerman, B.G. Faria, et al., "Internet-based solutions in the development and operation of an unmanned robotic airship", In *Proc. of the IEEE*, vol. 91 no. 3, pp. 463 – 474, March 2003.

[9] S. B. V. Gomes and J. Jr. G. Ramos, "Airship Dynamic Modeling for Autonomous Operation," *IEEE Int. Conf. on Robotics and Automation*, 1998, pp3462-3457

[10] E. Hygounenc, I. K. Jung, P. Soueres, S. Lacroix, "The autonomous blimp project at LAAS/CNRS: achievements in flight control and terrain mapping", in *Int. J. of Robotics Research*, vol. 23, no. 4-5, pp.473-511, April-May, 2004.

[11] C.S. Park, H. Lee, M. J. Tahk, H. Bang, "Airship Control Using Neural Network Augmented Model Inversion". *Proceedings of 2003 IEEE Conference on Control Applications*. Vol. 1, no.23-25 Jun. 2003 pp558 – 563

[12] J.R.Azinheira, P. E. Carneiro, J.G. Ramos and S.S. Beuno, "Mission path following for an autonomous unmanned airship", in *IEEE Int. Conf. on Robotics and Automation (ICRA2000).* Vol.2, no.24-28 pp1269 – 1275,April 2000

[13] J. G. Ramos, E. C. Paiva, J. R. Azinheira, S. S. Bueno, S. M. Maeta, L. G. B. Mirisola, et al. "Autonomous flight experiment with a robotic unmanned airship", in *Proc. 2001 IEEE Int. Conf. on Robotics & Automation*, Seoul, Korea, May, 2001.

[14] S.M. Smith, G.J.S. Rae, D.T. Anderson, "Applications of fuzzy logic to the control of an autonomous underwater vehicle". In *2nd IEEE Int. Conf. on Fuzzy Systems*. vol.2, pp1099–1106.

[15] A. Oyama, S. Obayashi, K. Nakahashi, "Real-coded adaptive range genetic algorithm and its application to aerodynamic design", *In JSME series A,* vol.43, no.2, 2000