Fuzzy-Genetic Autopilot Design for Nonminimum Phase and Nonlinear Unmanned Aerial Vehicles

Ali Reza Babaei1; Mahdi Mortazavi2; and Mohammad Hassan Moradi3

Abstract: In this paper, the authors tried to design the altitude hold mode autopilot for unmanned aerial vehicles. This case presents an interesting challenge attributable to the nonminimum phase characteristic, nonlinearities, and uncertainties of the altitude to elevator relationship. A fuzzy logic autopilot in a single-loop scheme is proposed for the design of this autopilot. The multiobjective genetic algorithm is used to mechanize the optimal determination of fuzzy logic autopilot parameters on the basis of an efficient cost function that comprises undershoot, overshoot, rise time, settling time, steady state error, and stability. Simulation results show that the proposed strategy not only has a simple structure, but also has desirable performances in time response characteristics, robustness (against the unmodeled nonlinear terms and parametric uncertainties), and the adaptation of itself than the large commands. DOI: 10.1061/(ASCE)AS.1943-5525.0000116. © 2012 American Society of Civil Engineers.

CE Database subject headings: Algorithms; Fuzzy sets; Uncertainty principles; Vehicles; Aerospace engineering.

Author keywords: Autopilot; Unmanned aerial vehicles; Genetic algorithm; Fuzzy logic; Uncertainty; Altitude hold mode.

Introduction

The main difficulties in aerospace vehicle autopilot design are attributable to aerodynamic parameter uncertainties. Because the knowledge of aerodynamic coefficients and their dependencies on some parameters (e.g., angle of attack and side slip angle) is very imprecise, a designed controller must not be sensitive to the variations of these coefficients. Also, aerospace vehicles are controlled by assuming that inertial cross couplings among roll, pitch, and yaw dynamics are negligible. In practice, the dynamics of aerospace vehicles is nonlinear, time varying, and uncertain. Nonlinear terms such as coupling among the dynamics, may often lead to performance degradation. To design a high performance autopilot, it is desirable to use a more general model containing the nonlinear terms. Therefore, one of the major problems in designing flight control systems is modeling uncertainties such as parameter variations and unknown nonlinear dynamics.

To get rid of the exact model restrictions, several adaptive schemes have been introduced to solve the problem of linearly parameterized uncertainties, which are referred to as structured uncertainties (Kanellakopoulos et al. 1991; Barkana 2005). In Barkana (2005), the altitude hold mode autopilot is designed to improve only parametric robustness. In this reference, unmanned aerial vehicles (UAVs) altitude is controlled by the combination of a classical controller and a simple adaptive controller on the basis of the linear model. Unfortunately, in some applications, some controlled systems characterized by an unmodeled or/and unknown dynamics are referred to as unstructured uncertainties. The feedback linearization (Slotine and Li 1991) is a popular method used in nonlinear control applications and several flight control demonstrations have presented this approach (Menon et al. 1987; Tahk et al. 1986; Azam and Singh 1994; Bugański and Enns 1992). This technique has received much of the attention and shows great promise. The main drawback to the nonlinear control approach such as feedback linearization is that as a model-based control method, they require accurate knowledge of the plant dynamics. This is significant in flight control because the aerodynamic parameters always contain some degree of uncertainty. In this paper, controller design methods are applied to the altitude hold mode autopilot for a UAV which is aerodynamically controlled by the elevator control surface, and the considered uncertainties are attributable to both parametric uncertainty and unmodeled nonlinear terms. In fact, the input/output feedback linearization technique, which has been regarded as a powerful design method for nonlinear systems, cannot be directly applied to the altitude control of UAVs because the altitude to the elevator relation is inherently nonminimum phase. These facts make it difficult to design a high performance autopilot.

The fuzzy logic control (FLC) method (Wang 1997) has been used for various cases (Wu et al. 2003; Kadmiry and Dríankov 2004; Cohen and Bossert 2003; Bossert and Cohen 2002) and it can be used as a robust method for our case. The optimal determination of fuzzy systems properties by using evolutionary algorithms (Haupt and Haupt 2004) has also been widely used in different applications (Wu and Tan 2006; Serra and Bottura 2006; Blumel et al. 2001; Li and Shieh 2000; Tsourds et al. 2006; Omar 2007). In Cohen and Bossert (2003), the altitude hold mode autopilot is designed on the basis of the linear model to improve only parametric robustness. In Cohen and Bossert (2003), the UAV altitude is controlled by using the expert knowledge-based FLC so that the robustness to parametric uncertainty is achieved. In Blumel et al. (2001) and Tsourds et al. (2006), the missile acceleration at nonminimum phase is indirectly controlled (lateral velocities are controlled) by FLC and a multiobjective genetic
algorithm (GA). In Li and Shieh (2000), the fuzzy proportional-integral-derivative (PID) controller on the basis of GA is presented to eliminate the undershoot of nonminimum phase linear systems. In Omar (2007), GA-based fuzzy logic controller has been designed to stabilize a satellite attitude.

In this paper, the authors tried to achieve some properties: suitable time response characteristics, robustness to parametric uncertainty, robustness to unmodeled dynamics, and desirable tracking of commands in the wide range. Here, unlike the conventional architecture, a single-loop scheme is used to design the altitude hold mode autopilot. In Barkana (2005) and Cohen and Bossert (2003), the single-loop scheme has been seen for the altitude autopilot. This architecture leads to decrease the required measurable variables (an advantage), but it also decreased robustness (a disadvantage). For this reason, a multiobjective GA-based FLC is proposed in the design of this autopilot. The FLC leads to achieving a robust autopilot because it is independent of the system model. Also, the optimal determination of FLC parameters can be mechanized by an efficient multiobjective GA.

The rest of the paper is organized as follows: first, the nonlinear and linear model of a UAV is presented. A linear autopilot is then designed and evaluated and the simulation results show degraded effects of coupling nonlinear terms that must be eliminated in the autopilot design procedure. Next, a description of a knowledge-based fuzzy logic autopilot is presented. Then, designing the GA-based FLC is presented and evaluated. Finally, conclusions are drawn.

UAV Model

Our purpose in this section is to derive the equations of motion of a UAV. The UAV model could be built from a set of simultaneous ODEs. The equations of motion of a UAV as a rigid body can be separated into rotational equations and translational equations. The rotational motion of a UAV will then be equivalent to yawing, pitching, and rolling motions about its center of mass. The remaining components of the motion will be translation of the center of mass in three-dimensional (3D) space. Therefore, the state model derived here will be a six-degree-of-freedom model.

We can derive the dynamics equations of a UAV as follows by the expansion of Newton laws (Blakelock 1991; Roskam 1990).

Translational dynamics equations

\[ F_x = m(\dot{U} + QW - RW) \]  
\[ F_y = m(\dot{V} + RU - PW) \]  
\[ F_z = m(\dot{W} + PV - QU) \]

Rotational dynamics equations

\[ L = I_x \dot{P} - I_z \dot{R} - I_{xy} PQ + (I_z - I_y) RQ \]  
\[ M = I_y \dot{Q} + (I_y - I_z) PR + I_{yz} (P^2 - R^2) \]  
\[ N = I_z \dot{R} - I_{yx} \dot{P} + (I_z - I_x) PQ + I_{xz} RQ \]

in which \( F_x, F_y, \) and \( F_z \) are aerodynamic forces in the \( x, y, \) and \( z \) axes; \( L, M, \) and \( N \) = aerodynamic and thrust moment about the \( x, y, \) and \( z \) axes; \( I_x, I_y, \) and \( I_z \) = moments of inertia; and \( m \) = mass of the UAV. By linear expansion of aerodynamic forces and moments and applying the gravity and thrust effects to these equations, the following equations are derived:

\[ \dot{U} = RV - QW - g \sin \theta + C_{\alpha U} U + C_{\alpha \alpha} \alpha 
+ C_{\alpha Q} Q + C_{\alpha E} E + F \Theta \]  
\[ \dot{V} = PW - RU + g \sin \phi \cos \theta + C_{\gamma \beta} \beta + C_{\gamma \rho} P 
+ C_{\gamma R} R + C_{\gamma \alpha} \delta_A + C_{\gamma \epsilon} \delta_R \]  
\[ \dot{W} = QU - PV + g \cos \phi \cos \theta + C_{\alpha U} U + C_{\alpha \alpha} \alpha 
+ C_{\alpha Q} Q + C_{\alpha E} E + F \Theta \]  
\[ \dot{P} = I_1 PQ + I_2 QR + C_{L \beta} \beta + C_{L \rho} P 
+ C_{L R} R + C_{L \alpha} \delta_A + C_{L \epsilon} \delta_R \]  
\[ \dot{Q} = I_3 PR + I_4 (R^2 - P^2) + C_{MU} U + C_{M \alpha} \alpha 
+ C_{M Q} Q + C_{M \epsilon} \delta_E + M_0 \]  
\[ \dot{R} = I_3 P Q - I_4 Q R + C_{N \beta} \beta + C_{N \rho} P 
+ C_{N R} R + C_{N \alpha} \delta_A + C_{N \epsilon} \delta_R \]

The basis of derivation of these equations is found in Blakelock (1991) and Roskam (1990). In these equations, the constants \( I_x, \) and \( C_{\alpha \alpha} \) are the functions of mass properties (mass and moment of inertia) and stability and control derivatives (these values are presented in Table 1). We have computed these derivatives by using procedures presented in Roskam (1990). Also, the kinematics equations are presented as follows (Rauw 2005):

\[ \dot{\alpha} = \frac{\dot{W} \cos \alpha - \dot{U} \sin \alpha}{V \cos \beta} \]  
\[ \dot{\beta} = \frac{1}{V \cos \beta} \left[ -\dot{U} \cos \alpha \sin \beta + \dot{V} \cos \beta - \dot{W} \sin \alpha \sin \beta \right] \]  
\[ \dot{\phi} = P + Q \sin \phi \tan \theta + R \cos \phi \tan \theta \]  
\[ \dot{\theta} = Q \cos \phi - R \sin \phi \]  
\[ \dot{\psi} = (Q \sin \phi + R \cos \phi) \sec \theta \]  
\[ \dot{h} = V \sin \gamma \]  
\[ V_t = \sqrt{U^2 + V^2 + W^2} \]  
\[ \gamma = \theta - \alpha \]

Clearly, Eqs. (7)–(20) are nonlinear and highly coupled. The aim of this study is to design the altitude hold mode autopilot for this UAV. The altitude hold mode autopilot is able to maintain the altitude by using the elevator input. This mode has practical importance for UAVs that fly in the vicinity of terrains to implement missions by terrain following maneuvering. This work has been motivated by the challenge of developing and implementing an autopilot that is robust to parametric uncertainties and unmodeled dynamics to implement efficiently a terrain following maneuver in a wide range of altitudes.

By the linearization of the nonlinear model of the UAV (the procedure of linearization of aircraft nonlinear model about steady state flight is presented in Blakelock (1991) and Roskam (1990)),
the nominal linear model for altitude is derived by using the following transfer function:

$$ h(s) = \frac{-57.3(s - 24.6)(s + 21)(s + 0.008)}{s(s^2 + 0.011s + 0.0022)(s^2 + 2.12s + 98.4)} $$

(21)

It can be received that the altitude output and the elevator input relation is in a nonminimum phase, and the longitudinal flight modes are not desirable (particularly, short period damping ratio and phugoid natural frequency).

If a 40% reduction is applied to $C_{MQ}$, $C_{Mo}$, $C_{U}$, and $C_{U}$ coefficients [see Table 2 (Blakelock 1991)], a degraded linear model is derived by using following transfer function:

$$ h(s) = \frac{-57.3(s - 25.5)(s + 21.6)(s + 0.0017)}{s(s^2 + 0.0055s + 0.0021)(s^2 + 1.82s + 64)} $$

(22)

These variations in the plant dynamics are caused by moving an unstable zero further to the right, and also caused the degradation of the phugoid and the short period flight modes.

In summary, the following points are discussed about the models of this UAV:

1. The nominal linear model is considered as an available mathematical model for the design of the autopilot.
2. The degraded linear model is used to investigate parametric robustness.
3. The nonlinear model as an actual model is used to investigate the robustness in unmodeled dynamics. Concerning Eqs. (7)–(20), the lateral-directional variables have an effect on the altitude variable. This means that there is a coupling between the lateral-directional channel and the longitudinal channel that has been canceled in the linearization procedure. Thus, it may be that the performance of designed autopilot on the basis of the nominal linear model is degraded after applying it to the nonlinear model.
4. In addition to the coupling nonlinear effects, the altitude and the elevator relationship is in nonminimum phase thus leading to the degradation of the autopilot performance.

### Linear Autopilot Design

Conventionally, the altitude autopilot is designed with three loops so that the altitude, pitch rate, and pitch angle variables are measured (the pitch angle autopilot is the inner loop of the altitude autopilot, and the pitch rate controller is the inner loop of the pitch angle autopilot). This architecture leads to a suitably robust yet complex and expensive controller because three controllers should be designed and three variables should be measured. In this paper, the pitch rate and the pitch angle loops are not included and thus, the single-loop architecture is used. In this simple architecture, it is sufficient to measure the altitude by using the altimeter.

The classic methods have the advantage of designing an autopilot with the capability of stability analysis and simplicity of implementation. These methods have a special place in automatic flight control systems. For these reasons, in this section, a classic autopilot is designed for $\psi$ and $\theta$ variables (a full-autopilot) by using the nominal linear model (a saturated linear dynamic is considered for the actuator). The $\psi$-autopilot that has the desirable performances is not presented and assumed that there are no uncertainties in the directional-lateral channel. The block diagram of the altitude hold mode for designing a compensator is shown in Fig. 1.

By considering a trade-off procedure between the time response characteristics and robustness, two different compensators are designed through the root locus techniques.

$$ G_{Cu} = \frac{0.00681(s + 0.1)(s^2 + 2.12s + 98.4)}{(s + 20)(s^2 + 6.94s + 13.1)} $$

(23)

$$ G_{Ca} = \frac{0.012(s + 0.05)(s^2 + 2.12s + 98.4)}{(s + 20)(s^2 + 6s + 15.25)} $$

(24)

By the analysis of the root locus and associated dominant pole, the closed loop system dominant characteristics are attained by applying these autopilots to the nominal linear model Eq. (21) as the following values:

$$ a : \xi = 0.79, \quad \omega_n = 1.36, \quad b : \xi = 0.6, \quad \omega_n = 2.31 $$

To evaluate these autopilots, they are applied to a nominal linear model, a degraded linear model, and a nonlinear model (the elevator trim angle $\delta_{E_{trim}} = 1.92^\circ$ must be added to the control input in the nonlinear simulation procedure because the controllers have been designed on the basis of the nominal linear model), then

---

**Table 2. Stability Derivative Effects on Dynamic Properties of UAVs**

<table>
<thead>
<tr>
<th>Stability derivative</th>
<th>Quantity most affected</th>
<th>How affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{MQ}$</td>
<td>Damping of short period, $\xi_s$</td>
<td>Increase $C_{MQ}$ to increase damping</td>
</tr>
<tr>
<td>$C_{Mo}$</td>
<td>Natural frequency of short period, $\omega_{\delta_{\beta}}$</td>
<td>Increase $C_{Mo}$ to increase frequency</td>
</tr>
<tr>
<td>$C_{U}$</td>
<td>Damping of phugoid, $\xi_p$</td>
<td>Increase $C_{U}$ to increase damping</td>
</tr>
<tr>
<td>$C_{U}$</td>
<td>Natural frequency of phugoid, $\omega_{\delta_{\beta}}$</td>
<td>Increase $C_{U}$ to increase frequency</td>
</tr>
</tbody>
</table>

---

**Fig. 1.** Single-loop scheme block diagram of the altitude hold mode autopilot to design a compensator

---

JOURNAL OF AEROSPACE ENGINEERING © ASCE / JANUARY 2012 / 3
the simulation results are investigated. The commands are considered as $\psi_c = 0$ and 10° and $h_c = 10$ m. The simulation results are illustrated in Figs. 2 and 3. According to Fig. 2, the desirable response is seen for the nominal linear model, but the relative parametric robustness of the first compensator and low parametric robustness of the second compensator is known. By inspection of Fig. 3, the time response of both classic autopilots has been degraded for the UAV nonlinear model because the nonlinear terms have appeared. Clearly, increasing the $\psi$-command value leads to increase the degraded effects of coupling nonlinear terms. Therefore, the compensators cannot meet the wanted requirements in parametric uncertainties and the unmodeled dynamics. Now, elimination of these degraded effects is required to achieve the desirable tracking.

In addition, the autopilot should not be very sensitive to the variation in system parameters (parametric robustness). Because of these reasons, this study tries to use knowledge-based FLC.

### Knowledge-Based Fuzzy Logic Autopilot

Fuzzy systems are knowledge or rule-based systems that were initiated by Zadeh (1965). The heart of a fuzzy system is a knowledge consisting of the so-called fuzzy if-then rules. A fuzzy if-then rule is an if-then statement in which some words are characterized by continuous membership functions. A block diagram of a fuzzy logic system is shown in Fig. 4.

In this paper, the fuzzy logic controller is implemented as a 17-rule Mamdani fuzzy system with two inputs and one output. The two inputs are error and error rate, and the output is the elevator deflection. Seven membership functions are used to describe each of inputs and output; namely, negative big, negative medium, negative small, zero, positive small, positive medium and positive big (Fig. 5). The rules base is presented in Table 3. These rules are justified by using phase plane and systems step response.
In first step, we use the relationship 3

functions variance.

Ne computed as

according to preceding relationship and values, the variances are

mathematical model of the UAV to achieve the requirements. Ac-

UA V flight characteristics and trial and error on the basis of the

problems. The GA is theoretically and empirically proven to

inspired by natural population genetics to generate solutions to

The GA is a general-purpose search algorithm that uses principles

of the UAV . Typically, first rule implies that

if e IS PB and ˙e IS ZE, then u IS PB (25)

Conventionally, FLC is designed by expert knowledge and experi-

ence. It is difficult to decide on control rules parameters as the

system gets complex, such as that of the nonminimum phase sys-

tem. It is difficult to decide on control rules parameters as the

functions variance.

Table 3. FLC Rules

<table>
<thead>
<tr>
<th>Rule number</th>
<th>e</th>
<th>˙e</th>
<th>u</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PB</td>
<td>ZE</td>
<td>PB</td>
</tr>
<tr>
<td>2</td>
<td>ZE</td>
<td>NB</td>
<td>NB</td>
</tr>
<tr>
<td>3</td>
<td>NB</td>
<td>ZE</td>
<td>NB</td>
</tr>
<tr>
<td>4</td>
<td>ZE</td>
<td>PB</td>
<td>PB</td>
</tr>
<tr>
<td>5</td>
<td>PM</td>
<td>ZE</td>
<td>PM</td>
</tr>
<tr>
<td>6</td>
<td>ZE</td>
<td>NM</td>
<td>NM</td>
</tr>
<tr>
<td>7</td>
<td>NM</td>
<td>ZE</td>
<td>NM</td>
</tr>
<tr>
<td>8</td>
<td>ZE</td>
<td>PM</td>
<td>PM</td>
</tr>
<tr>
<td>9</td>
<td>PS</td>
<td>ZE</td>
<td>PS</td>
</tr>
<tr>
<td>10</td>
<td>ZE</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>11</td>
<td>NS</td>
<td>ZE</td>
<td>NS</td>
</tr>
<tr>
<td>12</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
</tr>
<tr>
<td>13</td>
<td>ZE</td>
<td>ZE</td>
<td>ZE</td>
</tr>
<tr>
<td>14</td>
<td>PB</td>
<td>NS</td>
<td>PM</td>
</tr>
<tr>
<td>15</td>
<td>PS</td>
<td>NB</td>
<td>NM</td>
</tr>
<tr>
<td>16</td>
<td>NB</td>
<td>PS</td>
<td>NM</td>
</tr>
<tr>
<td>17</td>
<td>NS</td>
<td>PB</td>
<td>PM</td>
</tr>
</tbody>
</table>

(Lee 1990). For example, rules 16–17 are used to decrease the

undershoot that is created because of nonminimum phase behavior

of the UAV. Typically, first rule implies that

provide a robust search in complex spaces, thereby it is a valid ap-

proach to problems requiring efficient and effective searches.

Here, the continuous or real-valued GA is used. First, a chromo-

some population (Npop) is randomly generated. Each chromosome

specifies a candidate solution of the optimization problem. The fit-

ness of all individuals in the optimization task is then evaluated by a

scalar cost function (fitness function). A cost function generates an

output from a set of input variables (a chromosome). The goal is to

modify the output in some desirable fashion by finding the appro-

priate values for the input variables. If the chromosome has Nvar

variables given by p1, p2, ..., pNvar, then the chromosome is written

as an Nvar element row vector. Then, it is the time to decide which

chromosomes in the initial population are fit enough to survive and

dpossibly generate offspring in the next generation. The Npop costs

and associated chromosomes are ranked from lowest to highest

cost. From the Npop chromosomes in a given generation, only the

top Nkeep are kept for mating (process of natural selection) and

the rest are discarded to make room for the new offspring. Sub-

sequently, one mother and one father in some random fashion are

selected. Each pair produces two offspring that contain traits from

each parent. A single offspring variable value, pnew, comes from a

combination of the two corresponding offspring variable values

pnew = b0pm + (1 − b0)pdn (26)

If care is not taken, GA can converge too quickly into one region

of the cost surface. If this area is in the region of the global mini-

mum, that is good. However, some functions have many local

minimums. If this tendency to converge quickly is not solved, the

local minimum rather than the global minimum is attained.

To avoid this problem, it is forced to explore other areas of the cost

surface by randomly introducing changes, or mutations, in some of

the variables. Most users of the continuous GA add a normally dis-

tributed random number to the variable selected for mutation

p′n = pn + σNn(0, 1) (27)

GA-Based Fuzzy Logic Autopilot

Genetic Algorithm

The GA is a general-purpose search algorithm that uses principles

inspired by natural population genetics to generate solutions to

problems. The GA is theoretically and empirically proven to

Fig. 6. Nominal and degraded linear model time response with the

knowledge-based fuzzy logic autopilot

<table>
<thead>
<tr>
<th>Time,s</th>
<th>Nominal Linear Model</th>
<th>Degraded Linear Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>45</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>55</td>
<td>55</td>
<td>55</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>65</td>
<td>65</td>
<td>65</td>
</tr>
<tr>
<td>70</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>75</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>85</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>90</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>95</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>105</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>110</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>115</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>120</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>125</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>130</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>135</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>140</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>145</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>150</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>155</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>160</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>165</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>170</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>175</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>180</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

JOURNAL OF AEROSPACE ENGINEERING © ASCE / JANUARY 2012 / 5
Cost Function

In a desirable design of FLC by using GA, the presentation of an efficient cost function is of great importance. Here, a cost function is presented at which the undershoot, overshoot, settling time, steady state error, and unstable behavior are decreased. Thus, this is a multiobjective problem. The multiobjective cost function is proposed as follows:

\[ J = W_1 S_1 + W_2 S_2 + W_3 S_3 \]  

(28)

in which \( W_i \) = corresponding weight (Fig. 7); the parameters in Fig. 7 are defined as follows: \( t_1 \) = time of first intersection between the altitude time response and the altitude command; \( t_2 \) = time of second intersection between the altitude time response and the altitude command; and \( T \) = final time of simulation. If the first intersection is not created, parameters \( t_1 \) and \( t_2 \) are considered to be of zero value. If the first intersection is created and the second intersection is not created, parameter \( t_2 \) is considered equal to \( t_1 \).

\[ S_1 = \int_{0}^{t_1} |h - h_c|dt: \text{this parameter covers the rise time and undershoot;} \]

\[ S_2 = \int_{t_1}^{t_2} |h - h_c|dt: \text{this parameter includes the overshoot value;} \]

\[ S_3 = \int_{t_2}^{T} |h - h_c|dt: \text{this parameter includes the settling time, steady state error, and unstable effects.} \]

For computation of the cost function corresponding to any fuzzy autopilot, the closed loop behavior (altitude) and desired behavior (command altitude) of the UAV in the period of simulation \( T \) are required. In this alternative form of the cost function, the computation of all objectives is not individually required, and we can easily consider several important objectives in the simple cost function.

Results

Because of the computational requirements, FLC is generally evolved off-line (by using GA and a mathematical model of the controlled process). For this reason, the nominal linear model, as an available mathematical model of UAV, is utilized to achieve fuzzy autopilot parameters. Then, by using the degraded linear model and the nonlinear model in the simulation procedure, the robustness of the controller to the uncertainties that are not considered in the GA is investigated. The optimization procedure of the fuzzy autopilot is illustrated in Fig. 8. Fig. 9 shows the simulation or implementation procedure of the optimal fuzzy autopilot.

Again, the input and output variables boundaries are considered as the knowledge-based fuzzy logic autopilot case. The centers of output membership functions \((\mu_i)\) and variances \((\sigma_i)\) of input membership functions are considered as a chromosome. The GA with the following properties is used to determine the FLC parameters (these properties are determined by trial and error on the basis of the nominal linear model):

- Chromosome population, \( N_{pop} = 50 \)
- Number of generations = 50
- Mutation rate = 2%
- \( N_{keep} = 50\% \)

Also, by considering a trade-off procedure between the time response characteristics and the robustness, the weights are chosen as \( W_1 = 1 \), \( W_2 = 1.2 \), and \( W_3 = 1 \). The GA results are presented in Table 4. Also, the mean fitness value in each generation is shown in Fig. 10.

The simulation results are illustrated in Fig. 11 for the nominal linear model, degraded linear model, and nonlinear model. Concerning Fig. 11(a), this autopilot gives the desirable time response characteristics (dotted line) and a good parametric robustness (solid line). The desirable time response is shown in Fig. 11(b) by eliminating the degraded effects of coupling nonlinear terms. Therefore, the robustness to unmodeled dynamics can be achieved for the fuzzy logic autopilot.
UAVs with terrain following maneuver have large altitude changes that cause large commands for the autopilot. For large commands, applying a filter or a dynamic model (called trajectory generator) is necessary to avoid large inputs and large load factors because it leads to construct a smooth command (trajectory) with sufficiently low curvature. Here, without using the trajectory generation that is a complex process, the acceptable load factor can be attained without saturation of input by proper selection of lower and upper bounds for fuzzy logic autopilot inputs and output.

According to Fig. 12, the elevator is saturated (the maximum elevator angle is considered 25°) and applied load factor is high for classic autopilot due to the large commands. But, fuzzy autopilot is caused that the time response adapts itself with large commands so that the elevator is not saturated and the applied load factor is acceptable. In this figure, the second compensator result is shown.

If we are going to apply the fuzzy logic autopilot hardware, then the optimization procedure should be done on the basis of the available mathematical model of an UAV.

### Table 4. Optimal Properties of GA/Nominal Linear Model-Based Fuzzy Logic Autopilot

<table>
<thead>
<tr>
<th>$\sigma_{e_{NB}}$</th>
<th>$\sigma_{e_{NM}}$</th>
<th>$\sigma_{e_{NS}}$</th>
<th>$\sigma_{e_{ZE}}$</th>
<th>$\sigma_{e_{PS}}$</th>
<th>$\sigma_{e_{PM}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{e_{NM}}$</td>
<td>$\sigma_{e_{NS}}$</td>
<td>$\sigma_{e_{ZE}}$</td>
<td>$\sigma_{e_{PS}}$</td>
<td>$\sigma_{e_{PM}}$</td>
<td>$\sigma_{e_{PB}}$</td>
</tr>
<tr>
<td>13.27</td>
<td>19.22</td>
<td>11.68</td>
<td>8.09</td>
<td>11.73</td>
<td>16.25</td>
</tr>
</tbody>
</table>

$u_{NM}$ $u_{NS}$ $u_{ZE}$ $u_{PS}$ $u_{PM}$

<table>
<thead>
<tr>
<th>$u_{NM}$</th>
<th>$u_{NS}$</th>
<th>$u_{ZE}$</th>
<th>$u_{PS}$</th>
<th>$u_{PM}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>−9.5</td>
<td>−5.09</td>
<td>0</td>
<td>4.67</td>
<td>5.9</td>
</tr>
</tbody>
</table>

Fig. 10. Mean fitness value for generations

Fig. 11. Linear and nonlinear models time response with GA-based fuzzy logic autopilot

Fig. 12. Nonlinear model time responses with large command

Fig. 13. Optimization procedure of fuzzy autopilot on the basis of the nonlinear model as an available mathematical model of an UAV

Fig. 14. Simulation/implementation procedure of the optimal fuzzy autopilot (nonlinear model-based)
The simulation results show that the classic compensators cannot meet the wanted requirements in parametric uncertainties and unmodeled nonlinear dynamics. For this UAV, the authors tried non-linear model as an available mathematical model, and the performances of the autopilot are investigated in the presence of the real UAV (see Figs. 13 and 14). Evidently, this autopilot gives better results than the previous autopilot that was generated on the basis of the nominal linear model. The GA results, on the basis of the nonlinear model, are presented in Table 5. Also, the simulation results are illustrated in Fig. 15 for the nonlinear model (in the presence of parametric uncertainty and in the absence of parametric uncertainty) with $\psi_x = 10^\circ$. Clearly, six degrees of freedom nonlinear simulation shows more elimination of the degraded coupling terms as compared to the nominal linear-based autopilot because in the nonlinear model-based GA, the coupling terms are present.

### Conclusions

In this paper, the controller design methods have been applied to the altitude hold mode autopilot for a UAV, which is a nonminimum phase, and its model includes both parametric uncertainties and unmodeled nonlinear dynamics. For this UAV, the authors tried to describe some properties, not only of a simple architecture, but also suitable time response characteristics, robustness to the uncertainties, and desirable tracking of commands in the wide range. The simulation results show that the classic compensators cannot meet the wanted requirements in parametric uncertainties and unmodeled dynamics. For these reasons, this study has proposed that the fuzzy logic autopilot is designed through the use of a multiobjective genetic algorithm. This strategy includes some advantages: (1) unlike the conventional architecture in the altitude autopilot design, the single-loop scheme has been used so that it leads to a decrease in the required measurements; (2) in the fuzzy autopilot design procedure, an alternative form of the cost function has been defined to determine optimal FLC; (3) because of the independency of fuzzy autopilot on the UAV model, the fuzzy autopilot outperforms the classic autopilots in robustness to parametric uncertainties and to unmodeled nonlinear dynamics, whereas both have the comparable time response characteristics for the nominal linear model; and (4) by proper selection of the lower and upper bounds for fuzzy logic autopilot input and output, the time response adapts itself to the large commands, whereas the compensator does not have this capability (this arises from constructing fuzzy logic rules on the basis of the bounded input and output defined through membership functions). Finally, the fuzzy logic autopilot optimized by the presented multiobjective GA is recommended to design other autopilot modes in the presence of uncertainties.

### Table 5. Optimal Properties of GA/Nonlinear Model-Based Fuzzy Logic Autopilot

| $\sigma_{e_{h\bar{a}}} | \sigma_{e_{h\bar{a}}} | \sigma_{e_{h\bar{a}}} | \sigma_{e_{h\bar{a}}} | \sigma_{e_{h\bar{a}}} | \sigma_{e_{h\bar{a}}}
|---|---|---|---|---|---|
| 13.14 | 8.97 | 10.53 | 15.48 | 10.95 | 13 | 12.36
| $\sigma_{e_{h\bar{a}}} | \sigma_{e_{h\bar{a}}} | \sigma_{e_{h\bar{a}}} | \sigma_{e_{h\bar{a}}} | \sigma_{e_{h\bar{a}}} | \sigma_{e_{h\bar{a}}}
| 10.36 | 13.8 | 12.43 | 13.2 | 10.8 | 10.06 | 12.17

### Fig. 15. Nonlinear model time response with GA/nonlinear model-based fuzzy logic autopilot

### Notation

The following symbols are used in this paper:

- $C_a = $ constant parameter for stability and control (or aerodynamics) derivatives;
- $e = $ between output and command ($e = h - h_t$);
- $h = $ UAV altitude;
- $h_c = $ command altitude;
- $L = $ constant parameter, function of inertial characteristics of a UAV;
- $N = $ number of rules;
- $N_{keep} = $ number of kept chromosomes for mating;
- $N_{e}(0, 1) = $ standard normal distribution (mean = 0 and variance = 1);
- $N_{pop} = $ number of chromosome in population;
- $N_{var} = $ number of chromosome variables;
- $P = $ roll angle rate;
- $Q = $ pitch angle rate;
- $R = $ heading angle rate;
- $U = $ forward velocity;
- $V = $ lateral velocity;
- $V_t = $ total velocity;
- $W = $ vertical velocity;
- $\alpha = $ angle of attack;
- $\beta = $ side slip angle;
- $\delta_\alpha = $ aileron angle;
- $\delta_E = $ elevator angle (upward motion is positive);
- $\delta_{E_{trim}} = $ elevator trim angle;
- $\delta_R = $ rudder angle;
- $\delta = $ pitch angle;
- $\xi = $ damping ratio;
- $\sigma = $ standard deviation of normal distribution;
- $\phi = $ roll angle;
- $\psi = $ heading angle; and
- $\omega = $ natural frequency.

### References


Evolutionary Multi-Criterion Optimization (EMO), Springer-Verlag, Berlin, 668–680.


Roskam, J. (1990). Airplane design part VI: Preliminary calculation of aerodynamic thrust and power characteristics, Design Analysis & Research, Lawrence, KS.


