JOURNAL OF AIRCRAFT
Vol. 43, No. 1, January–February 2006

Flight Envelope Expansion in Landing Phase Using Classic, Intelligent, and Adaptive Controllers

S. M. B. Malaek* and Hojjat A. Izadi†
Sharif University of Technology, 11365 Tehran, Iran
and
Mehrdad Pakmehr‡
Concordia University, Montreal, Quebec H3G 1M8, Canada

An expanding flight envelope in the landing phase of a typical jet transport aircraft in presence of strong wind shears using a learning capable control system (LCCS) is investigated. The idea stems from human beings functional architecture that gives them the ability to do more as they age and gain more experience. With the knowledge that classical controllers lack sufficient generality to cope with nonlinear as well as uncertain phenomenon such as turbulent air, the focuse is on different types of intelligent controllers due to their learning and nonlinear generalization capabilities as candidates for the landing flight phase. It is shown that the latter class of controllers could be used to enable the aircraft to learn and gain experience as they land more and more in gusty conditions. This capability enables the aircraft to land in conditions slightly more severe than previous experiences, and hence, as far as the landing phase is concerned, the flight envelop is expanded as the aircraft ages. In this regard, five controllers including a classical proportional–integral–derivative (PID) plus three intelligent controllers, in addition to an adaptive one, are presented and compared, based on neural networks and fuzzy logic concepts. The robustness and performance of the controllers are evaluated by applying two different strong wind patterns for two levels of performance.

Nomenclature

A, B, C	=	system matrices
e	=	error function
g	=	gravity, 9.81 m/s^2
h	=	altitude, ft
h_{0f}	=	flare initiation altitude, ft
h_{0g}	=	glide initiation altitude, ft
K_*	=	controller's gain
M_*	=	pitching stability derivatives
q	=	pitch rate, deg/s
U_0	=	normal speed, 235 ft/s
u	=	longitudinal body-axis velocity, ft/s
u_g	=	longitudinal wind velocity, ft/s
w	=	vertical body-axis velocity, ft/s
w_g	=	vertical wind velocity, ft/s
X_*	=	forward stability derivatives
x	=	horizontal position of aircraft, ft
Z_*	=	upward stability derivatives
α	=	angle of attack, deg
α_s	=	stall angle of attack, deg
γ_0	=	flight-path angle, -3 deg
δ_E	=	elevator angle setting, deg
δ_T	=	throttle setting, deg
θ	=	pitch angle, deg
θ_p	=	controller threshold
μ	=	membership function
ω_*	=	controller's gain

Subscripts

c	=	desired amount of parameter
E	=	elevator setting
g	=	gust (wind shear)
T	=	throttle setting
TD	=	touchdown

Received 4 September 2004; revision received 24 January 2005; accepted for publication 25 January 2005.

*Associate Professor of Aerospace Engineering, P.O. Box 11365-8639, Aerospace Engineering Department, Azadi Avenue; malaek@sharif.edu.

†Graduate Student of Aerospace Engineering, P.O. Box 11365-8639, Aerospace Engineering Department, Azadi Avenue; ha_izadi@students.concordia.ca.

‡Graduate Student of Aerospace Engineering, Mechanical and Industrial Engineering Department, 1455 de Maisonneuve Boulevard; m_pakmeh@alcor.concordia.ca. Student Member AIAA.

I. Introduction

A FLIGHT safety envelope of an airplane indicates the regions that an airplane can safely fly without any undue risk and with an acceptable margin of safety. In the last decades, transport aircraft designers have adopted many innovative techniques such as enhanced vision systems to expand the flight envelope of airplanes and to increase the aircraft mission readiness while maintaining the safety of flight.

One of the most crucial phases of flight, as seen in a number of mishaps in the past decades, is the landing phase of flight. According to some published statistics, close to 60% of the flight-mishaps occur in the landing phase.[1] Other investigations confirm that, for a typical fleet of aircraft, landing normally has the highest percentage of accidents and/or incidents and, therefore, is considered as the most crucial and risky phase of a mission.[2]

Contributors to accidents could be categorized into two different sets. The first set is related to sensing errors, either human or sensor errors, such as altitude estimate error, runway conditions, and orientations. The second is due to sudden changes in atmospheric conditions. A typical example of such phenomenon, shown in Fig. 1, occurred at John F. Kennedy Airport in 1975 (Ref. 3). Similar gust and wind shear conditions are responsible for a high number of hard landings and mishaps each year.[4] This work focuses on the latter category, through developing a flexible human knowledge-based controller, which has the ability to adapt its performance based on changes in flight condition during landing.

Traditionally, classic controllers, such as proportional–integral–derivative (PID) controllers, have been used to control the aircraft

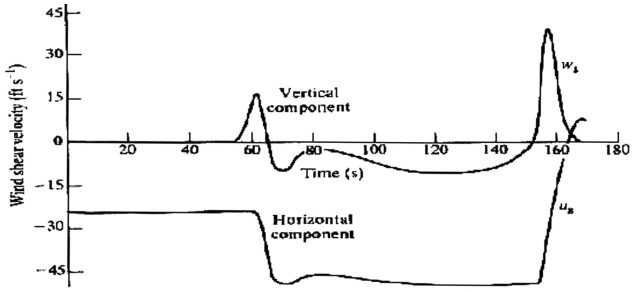

Fig. 1 JFK Airport downburst.[3]

through out the landing phase of a flight. However, these types of controllers need precise information about system dynamics and are sensitive to any changes in flight condition making it considerably hard to achieve an acceptable performance in a wide range of conditions. In fact, they have an effective capability just around the design point. Gain scheduling techniques have been used to solve this problem; however, it is necessary to implement an appropriate switching between gains, which is not possible in quick changing flight conditions such as landing in turbulent air. The objective of this paper is to propose a basic controller design with performance and stability during landing phase of flight in the presence of strong wind shears. Many research activities have been conducted on designing an automatic landing controller for different classes of aircraft, especially heavy jet transports. For example, in Ref. 5, an automatic landing system (ALS) based on a human skill model is described. The model is expressed as a nonlinear I/O mapping from the aircraft state to the control command provided by a human expert; a gain adaptation technique has also been introduced for robustness. In Ref. 6, a mixed H_2/H_∞ control technique has been employed to develop controllers for the ALS of a commercial airplane. In Ref. 7, H_∞ synthesis has been applied to the problem of designing a flare mode for automatic landing of a typical transport airplane. In Ref. 8, linear quadratic Gaussian with loop transfer recovery has been used to design an automatic landing controller for a typical commercial aircraft encountering a wind shear. In Ref. 9, adaptive critic-based neural networks have been used to design a controller for a benchmark problem in aircraft autolanding. In Ref. 10, by means of optimal control theory, seven neural networks were trained to learn the costates of the system and estimate the costates in similar scenarios without using the final time value to avoid solving Riccatti equation backward in time.

Different case studies conducted by researchers have resulted in a point that, although it would not be necessary to solve the Riccatti equation in the presence of gusts, the optimal-based neurocontroller does not normally have reasonable robustness. In Ref. 11, five different types of neural network structures are used to design intelligent autolanding controllers using linearized inverse dynamic modes; researchers tried to show how the type and complexity of a neural network is effective for landing flight phase. In Ref. 12, an adaptive controller based on model reference adaptive system (MRAS) methodology has been designed for a flight vehicle that enables it to track a predetermined flight-path trajectory in the presence of strong wind shears.

The outlined works lack the generality for a flight phase as crucial as the landing phase is. Landing as a flight phase varies considerably due to its vicinity to the ground, the existence of unknown patterns of wind and gust during the year, and natural obstacles and buildings and/or towers surrounding the airport. It is, therefore, desirable to develop a control system that can handle different climatic and situational conditions. Intelligent and adaptive-based control systems have been considered in this work to form a more robust controller for a landing flight phase.

A simple neurocontroller completely inherits the merits and weaknesses of its teacher architecture and works no better than the teacher controller.[13] Therefore, an intelligent controller of this type would not have sufficient generality for all forms of landing. To resolve this issue, an innovative hybrid neuro–PID controller was developed[14] that performs better than other controllers based on classic or intelligent methods. To achieve a still better performance, a fuzzy-based adaptive architecture was used in Ref. 15, and eventually, it was concluded that a hybrid fuzzy–PID controller fulfills all of the desired requirements. However, the latter architecture is hard to train.

In this paper, classical, neural-based, fuzzy-based, and adaptive controllers are compared to show the merits and weaknesses of each, and finally, the best architecture for a landing controller will be proposed. The procedure has six major steps as follows.

1) A PID controller is designed for a well-designed trajectory, with aircraft dynamics linearized in the vertical plane.

2) A neurocontroller is designed to control the aircraft throughout the glide and flare modes.

3) A hybrid neuro–PID controller is designed to handle the aircraft in a known very strong wind pattern. In this case, the inner loop is PID based and the outer loop is a neural-based controller.

4) A hybrid adaptive network based fuzzy inference system–(ANFIS–)controller is designed in which the inner loop is PID based and the outer loop is an ANFIS-based controller.

5) An adaptive controller by means of MRAS is designed for two purposes: first, to ease the training procedure and second, to provide a comparison benchmark as far as performance and robustness of the controller is concerned.

6) All designed controllers are simulated and compared for two different strong shear wind patterns.

In the current approach, the data and outputs generated by the PID controller are used to train intelligent controllers. Although, the PID controller is designed only for a single known trajectory, the new controller is expected to perform well in a wide range of flight conditions based on the characteristics of intelligent controllers.

II. Problem Formulation

A. Problem Definition and Objectives

Two major modes of the landing phase studied here are glide-slope hold–intercept and flare–touchdown with regard to definition given in Ref. 16. Normally, the glide mode begins around 500 ft above ground level (AGL) and ends around 45 ft AGL, which is the starting point for the flare mode (Fig. 2).

The flare mode continues until a smooth touchdown is achieved. During the glide-slope phase, the autolanding controller guides the aircraft along a straight line with a constant glide angle γ. The autopilot also attempts to prevent any changes in aircraft vertical and horizontal speeds and to hold them constant, that is, during glide mode the sink rate is constant.

As the flare phase begins, the autopilot starts to nose up the aircraft by changing the glide angle to prepare the aircraft for a smooth touchdown. The trajectory of aircraft during this mode is estimated by an exponential function. Through this process, the sink rate is reduced to a desired value of 1.5 ft/s, ensuring a safe and smooth touchdown.

Assuming the elevator and the throttle (used to control thrust) are the two controls during these two modes, we define two levels of performance known as level 1 and level 2, as follows.

Level 1 is the condition in which controllers should satisfy the following while flying in an expected gust: 1) $|\dot{h}| \leq 17$ ft/s and 2) $|\alpha| \leq 7$ deg.

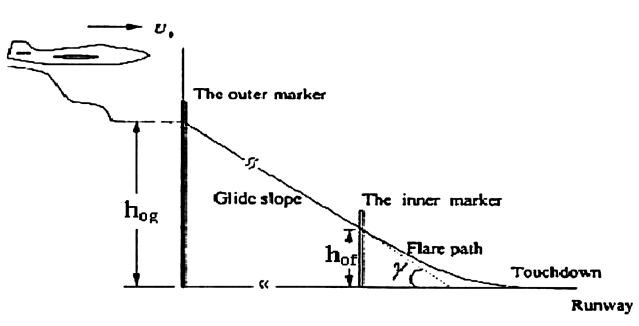

Fig. 2 Typical trajectory in landing phase.

Level 2 is the condition in which controllers should satisfy the following while flying in an expected gust: 1) $|\dot{h}| \le 20$ ft/s and 2) $|\alpha| \le 10$ deg.

Although, level 2 is acceptable, level 1 performance is preferred. These two levels of performance have been defined based on experimental data obtained through the actual flight tests of some aircrafts in the landing phase.[17,18]

B. Linearized Aircraft Equations of Motion

The model used here is a three-degree-of-freedom (longitudinal) rigid-body equation of motion with small perturbation theory, rendering linearized equations of motion. Therefore, the variation of all perturbed parameters must remain in some specified ranges as dictated by appropriate rules and regulations.[16]

The equations of motion at an altitude of 500 ft AGL with a trimmed speed of 235 ft/s are

$$\dot{u} = X_u(u-u_g) + X_w(w-w_g) + X_q q - g(\pi/180)\cos(\gamma_0)\theta + X_E\delta_E + X_T\delta_T \tag{1}$$

$$\dot{w} = Z_u(u-u_g) + Z_w(w-w_g) + [Z_q - (\pi/180)U_0]q + g(\pi/180)\sin(\gamma_0)\theta + Z_E\delta_E + Z_T\delta_T \tag{2}$$

$$\dot{q} = M_u(u-u_g) + M_w(w-w_g) + M_q q + M_E\delta_E + M_T\delta_T \tag{3}$$

$$\dot{\theta} = q \tag{4}$$

$$\dot{x} = u\cos\theta + w\sin\theta \tag{5}$$

$$\dot{h} = u\sin\theta - w\cos\theta \tag{6}$$

where the parameters X_*, Z_*, and M_* are stability and control derivatives of aircraft.

The initial conditions are assumed to be

$$u(0) = w(0) = q(0) = \theta(0) = 0 \tag{7}$$

$$h(0) = 500\,\text{ft}, \qquad x(0) = h(0)/\tan\gamma_0 \tag{8}$$

$$\dot{x}(0) = U_0 = 235\,\text{ft/s} \tag{9}$$

C. Turbulence Model

A random model of wind shear has been developed, which is used to evaluate the performance of the aircraft. The model consists of two parts: the constant part (u_{gc}, 0) and a variable part (u_{g1}, w_g). We assume the constant velocity component exists in the horizontal direction only and is given by

$$u_{gc} = \begin{cases} -u_0(1 + \ell_n(h/510)/\ell_n 51), & h \ge 10 \quad (10a) \\ 0, & h < 10 \quad (10b) \end{cases}$$

where u_0 is the wind speed at altitude 510 ft and typically has magnitude of 20 ft/s. The variable part (turbulence) is given by

$$u_g = u_{g1} + u_{gc} \tag{11}$$

$$w_g = \sigma_w\sqrt{a_w}\left(a_w w_{g1} + \sqrt{3}w_{g2}\right) \tag{12}$$

$$\dot{u}_{g1} = 0.2|u_{gc}|\sqrt{2a_u}N_1 - a_u u_{g1} \tag{13}$$

$$\dot{w}_{g1} = w_{g2} \tag{14}$$

$$\dot{w}_{g2} = N_2 - a_w^2 w_{g1} - 2a_w w_{g2} \tag{15}$$

where

$$a_u = \begin{cases} U_0\big/\left(100^3\sqrt{h}\right), & h > 230 \quad (16a) \\ U_0/600, & h \le 230 \quad (16b) \end{cases}$$

$$a_w = U_0/h \tag{17}$$

$$\sigma_w = \begin{cases} 0.2|u_{gc}|, & h > 500 \quad (18a) \\ 0.2|u_{gc}|(0.5 + 0.00098h), & h \le 500 \quad (18b) \end{cases}$$

In Eqs. (13) and (15), N_1 and N_2 are the Gaussian random noises with zero mean and different variances. Using Eqs. (10–18), one can generate wind patterns with different velocities and intensities.

III. Longitudinal Autolanding Controller Design

As already mentioned, a conventional PID, a neurocontroller, a hybrid neuro–PID, an ANFIS–PID, and an adaptive controller have been designed, which are now briefly described.

A. Basics of PID Controller

Two purposes are considered in the design of PID controller: 1) evaluation of this type of controller during landing, compared to other controllers and 2) generation of output data to train the neurocontroller.

The PID controller gains are estimated by application of the linear matrix inequality (LMI) method, which is normally used to design PID controllers for multi-input/multi-output (MIMO) systems. This method guarantees the stability of the designed system[19]; nevertheless, to achieve the desired performance, one needs to optimize the gains through a trial and error process.

Two effectors (controls) are commanded: throttle and elevator. Throttle is used to maintain constant aircraft speed during the landing phase.[5] The PID controller for controlling speed is

$$T = K_T(u_c - u) + K_T\omega_T\int_0^t (u_c - u)\,\mathrm{d}t \tag{19}$$

In this case, $u_c = 0$, $k_T = 3$, and $w_T = 0.1$.

The function of the elevator is to control both pitch angle and pitch rate during the landing phase; thus,

$$\delta_E = K_\theta(\theta_c - \theta) - K_q q \tag{20}$$

It is further assumed that the desired pitch angle is a function of error in h and $\dot{h}$,

$$\theta_c = k_h(h_c - h) + k_h\omega_h\int_0^t (h_c - h)\,\mathrm{d}t + k_{\dot{h}}(\dot{h}_c - \dot{h}) + \theta_p \tag{21}$$

where, for the current case,

$$K_h = 0.3, \qquad W_h = 0.1, \qquad K_{\dot{h}} = 0.3$$

At the glide mode,

$$K_\theta = 3, \qquad K_q = 3, \qquad \theta_p = 0$$

At the flare mode,

$$K_\theta = 12, \qquad K_q = 6.0, \qquad \theta_p = 0.0698$$

For each mode h_c and $\dot{h}_c$ are obtained from their trajectories in the glide and flare modes as the aircraft moves along a constant slope path characterized by

$$\tan\gamma_0 = h_c/x \Rightarrow h_c = x\tan\gamma_0 \tag{22}$$

where x could be measured with some appropriate means of navigation such as a global positioning system. During the flare mode, the controller is applied to the aircraft so that $\dot{h}$ is reduced smoothly to a desired value of -1.5 ft/s. When it is assumed that the flare mode begins at $t = t_0$ and $h_{0f} = h(t_0) = 45$ ft and it ends at the main gear touchdown point where $t = T$ and $h(T) = 0$, then

$$h_c = [h_0/(\dot{h}_0 - \dot{h}_{\text{TD}})]\left[\dot{h}_0 e^{-(x-x_0)\tau} - \dot{h}_{\text{TD}}\right] \tag{23}$$

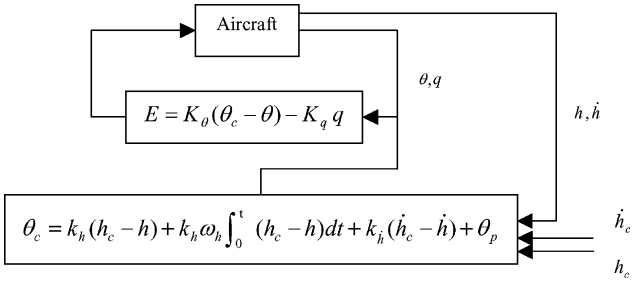

Fig. 3 PID controller block diagram.

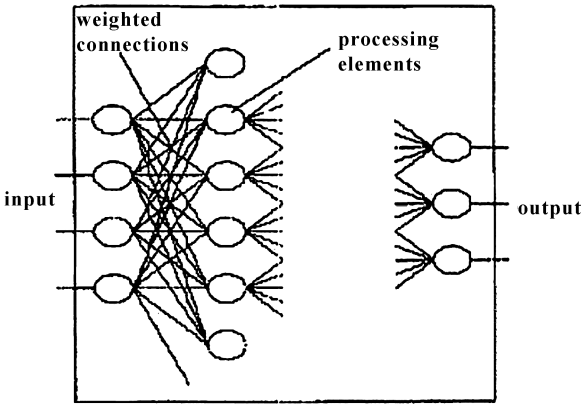

Fig. 4 Basic structure of artificial neural network.

where

$$\dot{h}_{\mathrm{TD}} = \dot{h}(\tau) = -1.5\ \mathrm{ft/s} \tag{24}$$

$$\tau = [-h_0\,\dot{x}(t_0)]/(\dot{h}_0 - \dot{h}_T) \tag{25}$$

Results of different simulations conducted by the authors show that setting throttle command to zero results in much better trajectories. Therefore, in the rest of the simulations, the throttle setting is treated as a constant. This helps to omit engine dynamics from simulation. It is well known that engine dynamics is rather slow compared to other controlling means and because during the landing condition the aircraft speed does not need to vary very much, the generality of the simulation would not be compromised. The block diagram of a PID controller is shown in Fig. 3.

B. Basics of Neurocontroller

An innovative methodology for modeling procedural and reflexive functions is artificial neural networks. They may be described as nonlinear multivariable function approximators and are motivated by I/O and learning properties of biological neural systems. Artificial neural networks consist of processing elements interconnected by weighted links that simulate the neurons and synapses of biological nervous systems (Fig. 4). Their knowledge is acquired by learning, which can be either supervised or unsupervised.[20]

The method presented here takes advantage of the fact that neural networks organize themselves while earning experience. In this work, one neural network is designed for elevator control based on a supervised training method.

As already mentioned, the outputs of PID controller are used to train the neural network. The neural network used for elevator control is a multilayer perceptron (MLP), called elevatornet, which has three layers and four inputs (θ, q, h, and $\dot{h}$). The output of elevatornet is the elevator command. The hidden layer has seven neurons ($N_{4,7,1}$). Moreover, a tangent-sigmoid function is used in the input, and hidden layers and pure linear function are used in output layers. To train the network, classical error backpropagation method (Levenberg–Marquardt backpropagation) is used. This method updates weight and bias values according to Levenberg–Marquardt optimization (see Ref. 21). The block diagram of this controller is shown in Fig. 5.

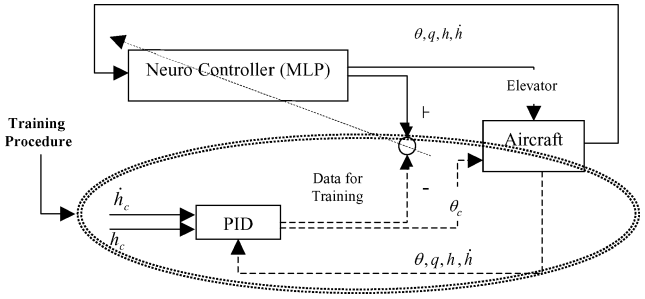

Fig. 5 Neurocontroller block diagram (with training procedure).

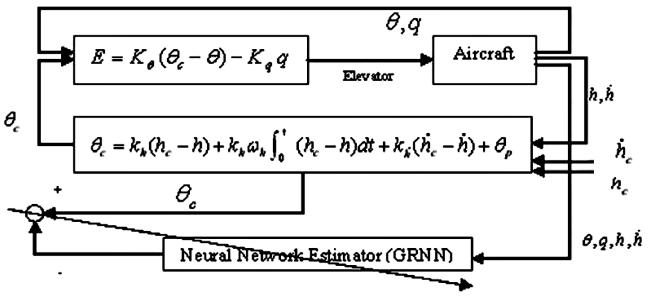

Fig. 6 Neuro-PID controller block diagram (with training procedure).

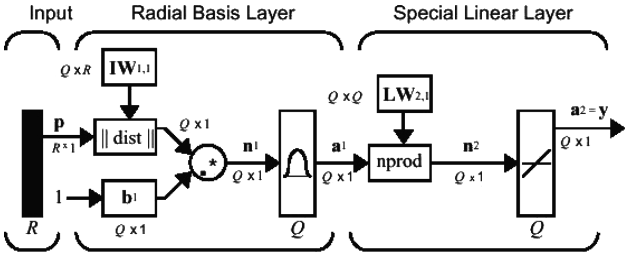

Fig. 7 Structure of GRNN[21]: where R = number of elements in input vector, Q = number of neurons in layer 1, Q = number of neurons in layer 2, Q = number of input/target pairs, $a_i1 = \mathrm{radbas}(\|_i1W_{1,1} - p\|b_i1)$, $a^2 = \mathrm{purelin}(n^2)$, a_i1 is ith element of a^1 where $_i1W_{1,1}$ is a vector made of the ith row of $1w_{1,1}$.

C. Basics of Hybrid Neuro–PID Controller

To enhance the aircraft landing performance in the presence of very strong winds and gusts, a new controller is proposed in which the inner loop provides the required stability for the system with the aid of classical methods (Fig. 6). In this controller, inner loop gains K_θ and K_q are set using a root locus plot of the system [Eq. (20)]. The outer-loop parameter θ_c is estimated by a type of neural network called general regression neural networks (GRNN). A GRNN is often used for function approximation. The architecture for the GRNN is shown in Fig. 7. As shown in Fig. 7, it has a radial basis layer and a special linear layer.[21]

D. Basics of Hybrid ANFIS–PID Controller Design

Similar to the neuro–PID controller, the hybrid ANFIS–PID controller is developed to further enhance the controller performance; however, in this case the outer loop is fuzzy based, which is discussed in this section.

1. ANFIS

ANFIS is mainly composed of a fuzzy section and an adaptive network section acting similar to a fuzzy system. However, it is structurally similar to an adaptive network (or perceptron neural networks). To show how ANFIS architecture works in the designed

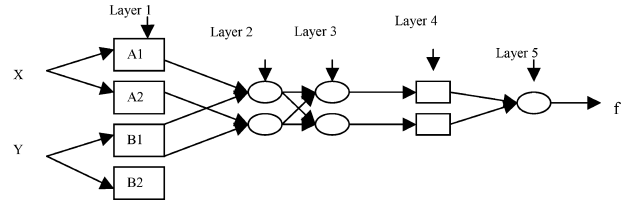

Fig. 8 ANFIS equivalent of Sugeno-type fuzzy system.

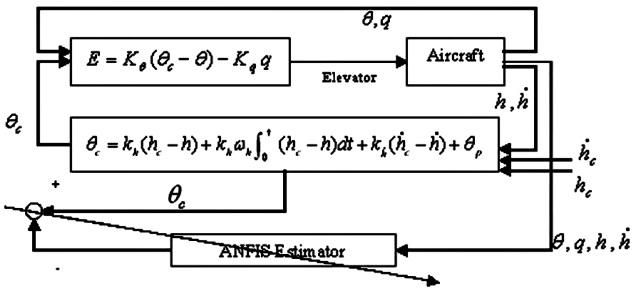

Fig. 9 ANFIS–PID Controller block diagram (with training procedure).

controller, consider the following Sugeno-type fuzzy system, in which $\boldsymbol{x}$ and $\boldsymbol{y}$ are the system inputs and $\boldsymbol{f}$ is its output.

Rule 1 is if $\boldsymbol{x}$ is $\boldsymbol{A}_1$ and $\boldsymbol{y}$ is $\boldsymbol{B}_1$, then $\boldsymbol{f}_1 = \boldsymbol{P}_1\boldsymbol{x} + \boldsymbol{Q}_1\boldsymbol{y} + \boldsymbol{r}_1$.

Rule 2 is if $\boldsymbol{x}$ is $\boldsymbol{A}_2$ and $\boldsymbol{y}$ is $\boldsymbol{B}_2$, then $\boldsymbol{f}_2 = \boldsymbol{P}_2\boldsymbol{x} + \boldsymbol{Q}_2\boldsymbol{y} + \boldsymbol{r}_2$ and

$$f = (w_1 f_1 + w_2 f_2)/w_1 \tag{26}$$

The equivalent ANFIS system for the preceding fuzzy system is shown in Fig. 8 (Ref. 22). Layer 1 is a membership layer in which the membership degree of $\boldsymbol{x}$ relative to $\mathbf{A}_i$ computes the output of layer 1 as follows:

$$O_i^{'} = \mu_{A_i}(x) \tag{27}$$

where μ is membership function of $\boldsymbol{A}_i$ and μ is a Gaussian (bell-shaped) function.

In layer 2, input signals are multiplied,

$$w_i = \mu_{A_i}(x)\mu_{B_i}(y) \tag{28}$$

That expresses the firing strength of each fuzzy rule. The output of each node in layer 3 is

$$\overline{w} = w_i/(w_1 + w_2) \tag{29}$$

which is called normalized firing strength. Each node in layer 4 is a square node with node function

$$O_2^4 = \bar{w}_i f_t - \bar{w}_z(P_z x + Q_z y + r_z) \tag{30}$$

Therefore, networks like Sugeno-type fuzzy systems could be provided by using the concept of adaptive (perceptron) networks.

The $\{\boldsymbol{P}_i, \boldsymbol{Q}_i, \boldsymbol{r}_i\}$ parameters and parameters of membership functions are defined by a training process, which is usually based on steepest descent method or least-square error method.

2. ANFIS–PID Controller Design

As already mentioned, in this controller the inner loop is designed by a classic PID tuning method and the outer loop is designed using ANFIS (Fig. 9). To train the ANFIS, two error parameters are defined as follows:

$$e_h = h_c - h, \qquad e_{\dot{h}} = \dot{h}_c - \dot{h} \tag{31}$$

These two parameters, e_h and $e_{\dot{h}}$, are inputs of the ANFIS network and the output is θ_c. In other words, ANFIS network, based on these two error values, e_h and $e_{\dot{h}}$, estimates the desired pitch angle θ_c for the purpose of compensation and correction. Each input, e_h and $e_{\dot{h}}$, has three bell-shaped membership functions. Linguistic variables for these membership functions are low, medium, and high. Consequently, after 20 epochs training of ANFIS, 9 fuzzy laws are generated as follows.[23]

1) If e_h is low and $e_{\dot{h}}$ is low, then θ_c is very low.
2) If e_h is low and $e_{\dot{h}}$ is medium, then θ_c is very low.
3) If e_h is low and $e_{\dot{h}}$ is high, then θ_c is low.
4) If e_h is medium and $e_{\dot{h}}$ is low, then θ_c is low medium.
5) If e_h is medium and $e_{\dot{h}}$ is medium, then θ_c is medium.
6) If e_h is medium and $e_{\dot{h}}$ is high, then θ_c is high medium.
7) If e_h is high and $e_{\dot{h}}$ is low, then θ_c is high.
8) If e_h is high and $e_{\dot{h}}$ is medium, then θ_c is very high.
9) If e_h is high and $e_{\dot{h}}$ is high, then θ_c is very very high.

The block diagram of the ANFIS–PID controller is shown in Fig. 9.

E. Basics of Adaptive Controller Design

In Fig. 10, the block diagram of MRAS architecture is shown. If the airplane model is specified by

$$\dot{X} = AX + BU, \qquad Y = CX \tag{32}$$

and ω is the vector of the controller, then using Lyapanov stability criteria (see Ref. 24)

$$\frac{\mathrm{d}\omega}{\mathrm{d}t} = -\gamma u_c B^T PX \tag{33}$$

That is, the controller is updated according to earlier adaptation law based on any change in the flight condition and/or airplane parameters. If the transfer function of system, $G(s)$, is strictly positive real (SPR), then, based on Ref. 24, one can write

$$B^T P = C \tag{34}$$

$$\frac{\mathrm{d}\omega}{\mathrm{d}t} = -\gamma u_c e \tag{35}$$

For the preceding system, the related transfer function is SPR if there exist positive definite matrixes P and Q so that

$$A^T P + PA = -Q \tag{36}$$

Now, let

$$\omega = [k_\theta \quad k_q]^T \tag{37}$$

$$y = [h, \dot{h}]^T \tag{38}$$

For the selected system here, Eq. (34) leads to a system of equations with 10 equations and 25 unknowns that can have infinite defined answers for P. Because A is a definite matrix, then from Eq. (36) there exists at least a definite solution for Q. Therefore, the aforementioned system is SPR and one can apply the adaptation law (35). The block diagram of this controller is shown in Fig. 10.

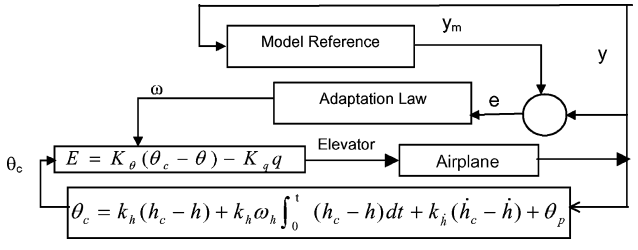

Fig. 10 Adaptive controller block diagram.

IV. Simulation Results

To train the networks, the Neural Networks Toolbox of MATLAB®[21] is used, and to simulate the control systems, Simulink[25] is used. Some proper links between the M-files and Simulink environment are necessary for the purpose of simulation. The initial conditions for all of the aforementioned controllers are introduced in Eqs. (7–9).

For numerical results, headwind magnitude is set to 25 kn (42.23 ft/s) and tailwind is set to 10 kn (16.9 ft/s) (Ref. 26). Figures 11–14 show the horizontal and vertical components of strong and very strong winds applied to the controllers. Simulation results are presented in two sections. In the first section, simulation results of the controllers in the presence of strong wind are given, and in the second section, the results in presence of very strong wind are presented. All controllers are tested in the presence of weaker winds in a wide range, which satisfies all necessary conditions. Because of space limitations, a limited number are provided and just the important cases are discussed.

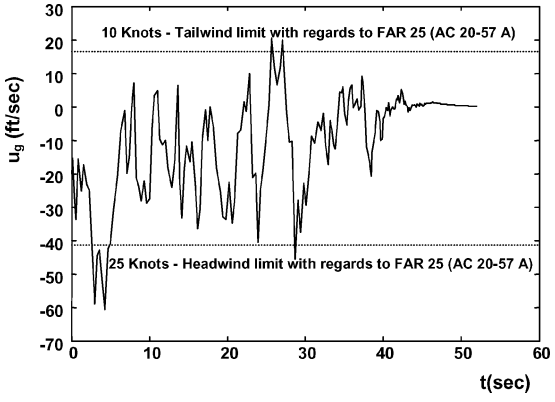

Fig. 11 Strong wind pattern, variation of u_g with h, $N = 100$.

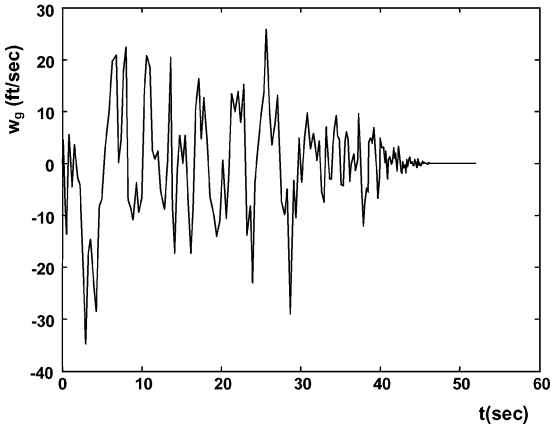

Fig. 12 Strong wind pattern, variation of w_g with h, $N = 100$.

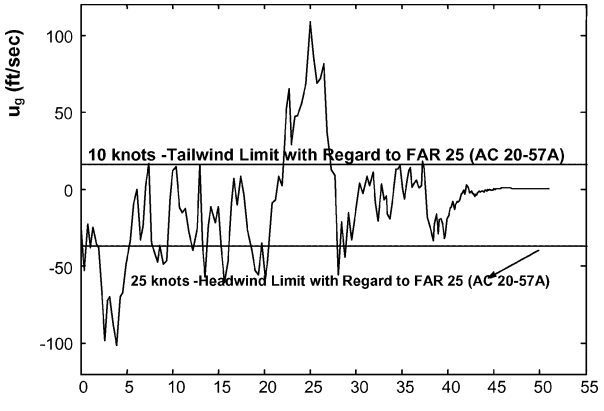

Fig. 13 Very strong wind pattern, variation of u_g with h, $N = 300$.

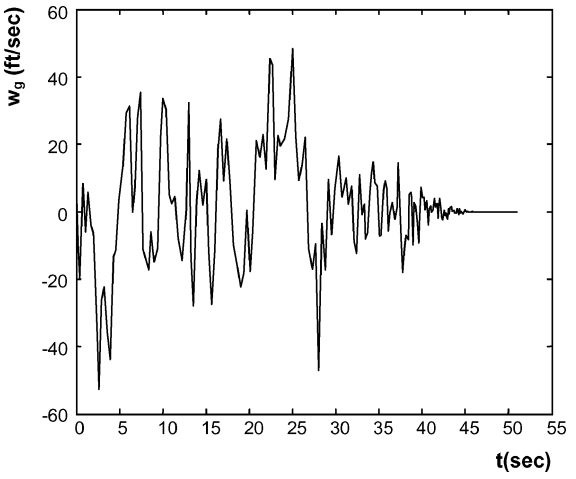

Fig. 14 Very strong wind pattern, variation of w_g with h, $N = 250$.

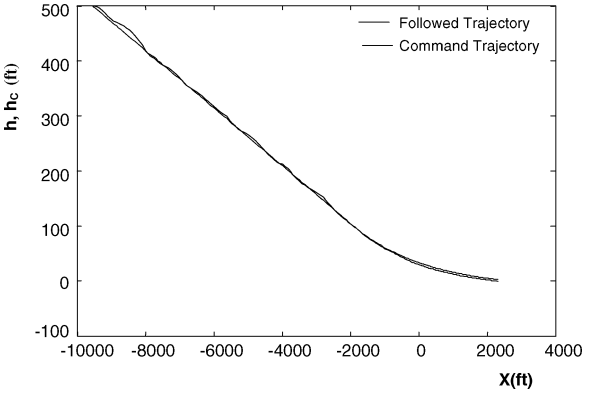

Fig. 15 Trajectory for PID controller with strong wind.

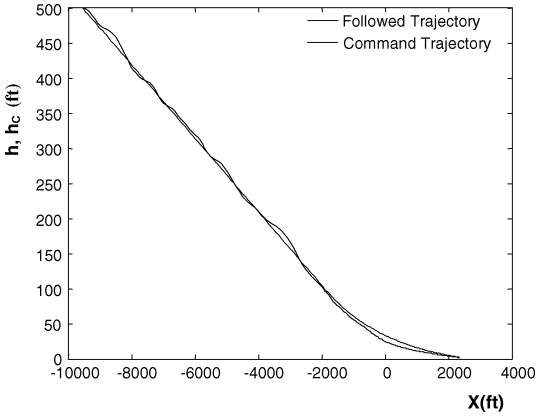

Fig. 16 Trajectory for neurocontroller with strong wind.

A. Results for Strong Wind

Simulation results for the strong wind (Figs. 11 and 12) are shown in Figs. 15–29. Actual and commanded trajectories of the aircraft for all of five controllers are shown in Figs. 15–19. As it is seen for all controllers, deviation from desired trajectory is not unexpected. The sink rates associated with each controller, shown in Figs. 20–24, satisfy the necessary conditions of level 2, whereas ANFIS and the adaptive controller go further and satisfy conditions of level 1. Variations of the aircraft angles of attack during the landing phase (Figs. 25–29) are all in an acceptable range compared to maximum selected stalling angle α_s of 5 deg (Ref. 16). It is further shown that all controllers have the same performance in the presence of a strong wind.

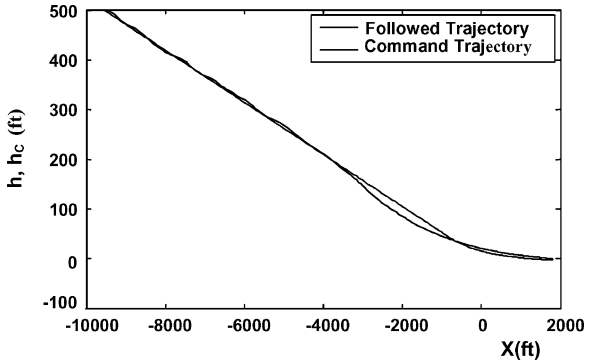

Fig. 17 Trajectory for neuro–PID controller with strong wind.

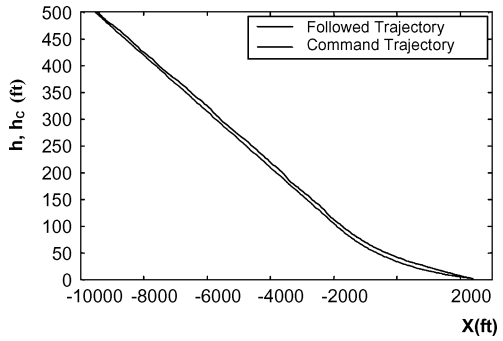

Fig. 18 Trajectory for ANFIS–PID controller with strong wind.

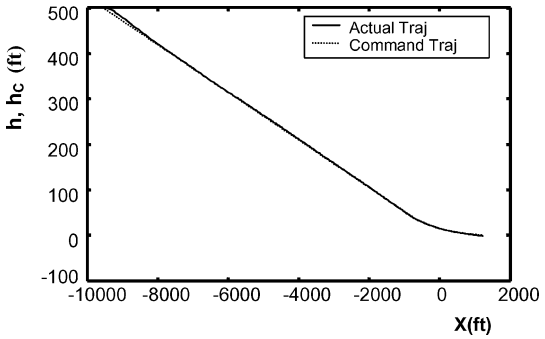

Fig. 19 Trajectory for adaptive controller with strong wind.

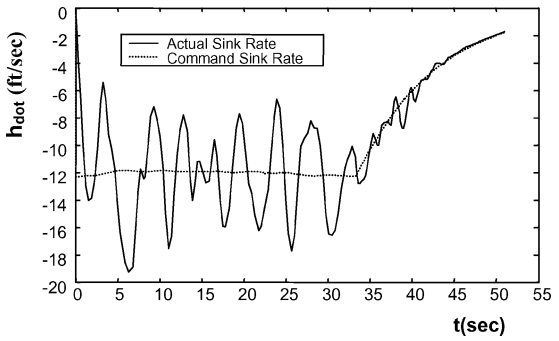

Fig. 20 Sink rate variations for PID controller with strong wind.

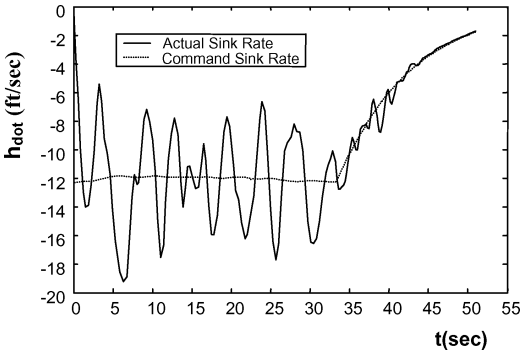

Fig. 21 Sink rate variations for neurocontroller with strong wind.

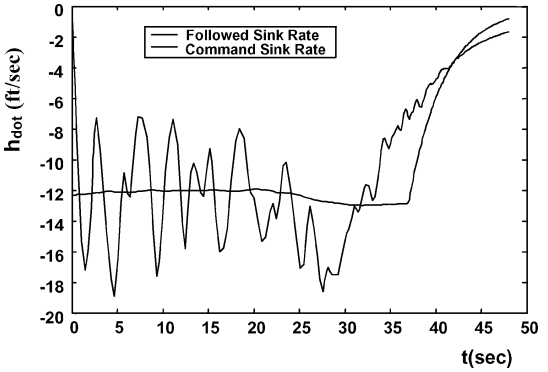

Fig. 22 Sink rate variations for neuro–PID controller with strong wind.

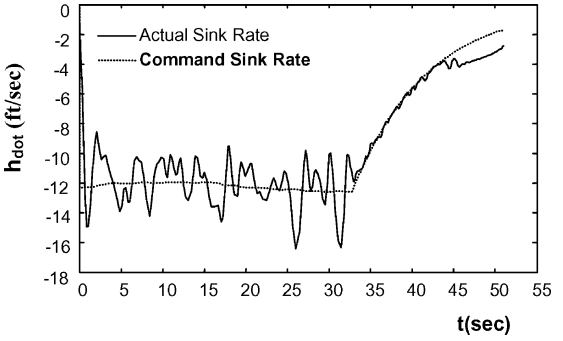

Fig. 23 Sink rate variations for ANFIS–PID controller with strong wind.

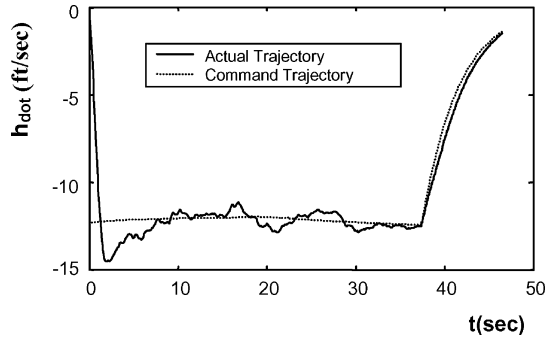

Fig. 24 Sink rate variations for adaptive controller with strong wind.

B. Results for Very Strong Wind

Figures 30–39 show simulation results of the controllers in presence of very strong wind (Figs. 13 and 14). Followed and command trajectories are shown in Figs. 30–34. Figure 30 shows that the classic controller follows the command trajectory in an acceptable manner, whereas the neurocontroller does not have the same performance (Fig. 31). However, Fig. 32 shows that the neuro–PID controller has a relatively better performance. In comparison, ANFIS and adaptive controllers have a much better performance (Figs. 33 and 34).

Desired and actual sink rates for the controllers are shown in Figs. 35–39. Actual sink rate of both classical and neurocontrollers (Figs. 35 and 36) exceed −20 ft/s and satisfy neither level 1 nor level 2 flying qualities. Sink rate of the aircraft with hybrid neuro–PID controller (Fig. 37) in the presence of very strong wind does not exceed the −20-ft/s limitation and satisfies the necessary conditions

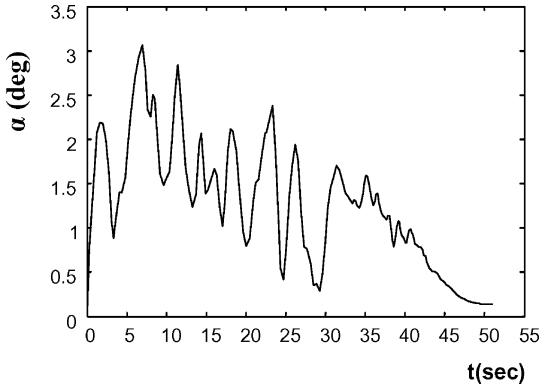

Fig. 25 Angle-of-attack variations for PID controller with strong wind.

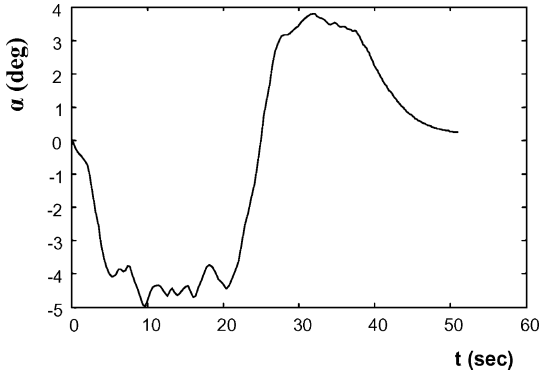

Fig. 26 Angle-of-attack variations for neurocontroller with strong wind.

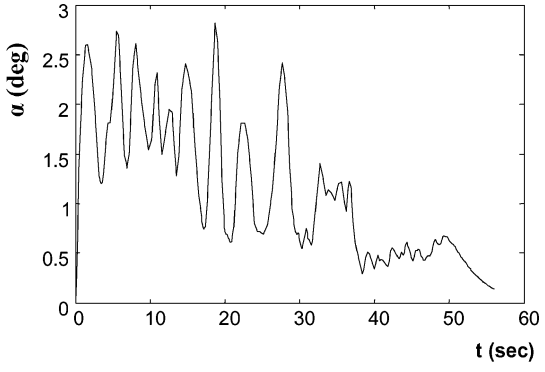

Fig. 27 Angle-of-attack variations for neuro–PID controller with strong wind.

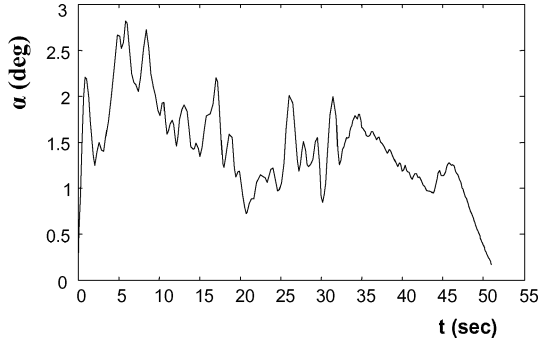

Fig. 28 Angle-of-attack variations for ANFIS–PID controller with strong wind.

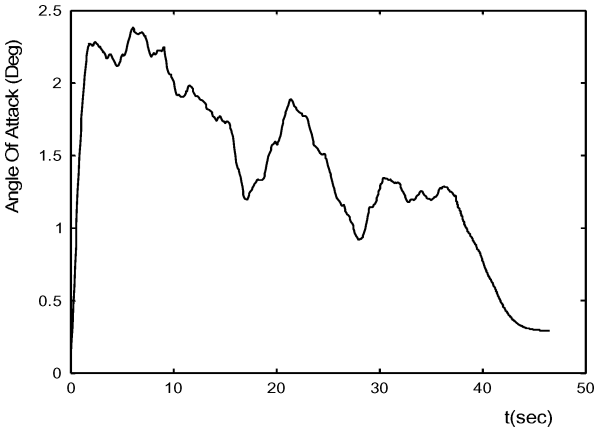

Fig. 29 Angle-of-attack variations for adaptive controller with strong wind.

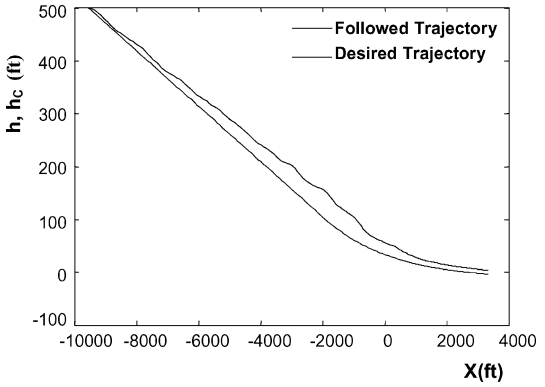

Fig. 30 Trajectory for PID controller with very strong wind.

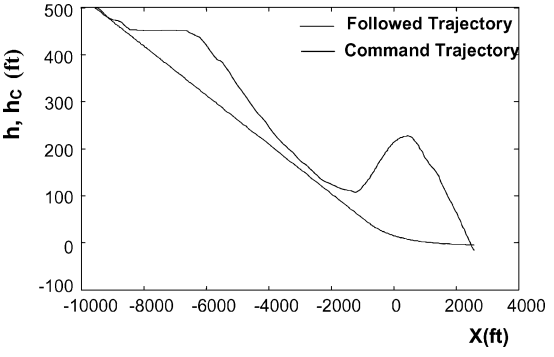

Fig. 31 Trajectory for neurocontroller with very strong wind.

for level 2. Figures 38 and 39 show that the actual sink rate with ANFIS–PID and adaptive controller in the presence of very strong wind does not exceed 17 ft/s and, hence, satisfies the necessary conditions for level 1. Moreover, ANFIS–PID and adaptive controllers render a relatively better performance for angle-of-attack variations. It is then concluded that ANFIS–PID and adaptive controllers have the potential to expand effectively the flight envelope of the aircraft by flying through very strong winds, with some additional cost for employing such controllers.

Table 1 summarizes the findings.

V. Complexity and Implementation Discussion

Complexity of the controllers could be discussed from four different aspects: 1) number of required sensors, 2) amount of required computations and calculations, 3) required amount of switching, and 4) implementation. Block diagrams of the PID, neuro-, neuro–PID, ANFIS–PID, and adaptive controllers are shown in Figs. 3, 5, 6, 9, and 10. As it is obvious, θ, q, h, and $\dot{h}$ are the inputs of all

Table 1 Performance of the controllers in presence of different wind patterns

Wind pattern/controller	PID	Neurocontroller	Neuro-PID	ANFIS–PID	Adaptive
Strong wind	Level 2	Level 2	Level 2	Level 1	Level 1
Very strong wind	Unacceptable	Unacceptable	Level 2	Level 1	Level 1

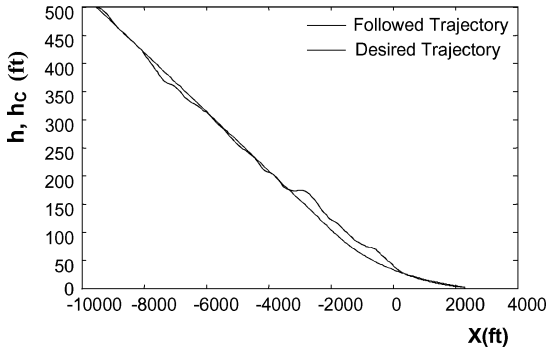

Fig. 32 Trajectory for neuro–PID controller with very strong wind.

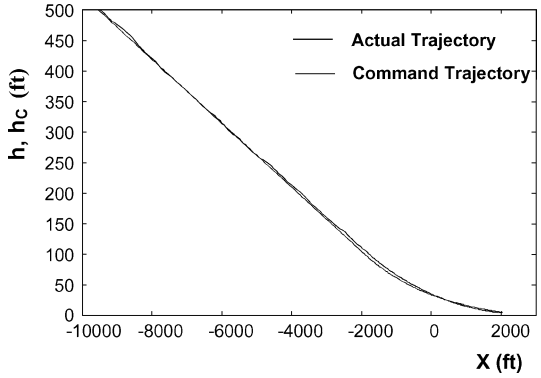

Fig. 33 Trajectory for ANFIS–PID controller with very strong wind.

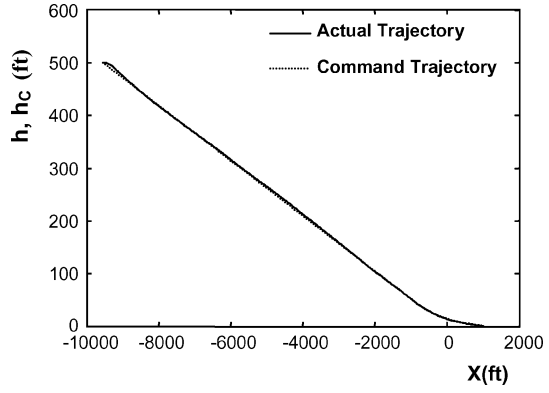

Fig. 34 Trajectory for adaptive controller with very strong wind.

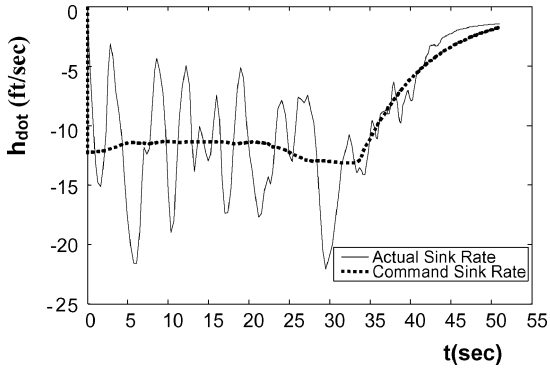

Fig. 35 Sink rate variations for PID controller with very strong wind.

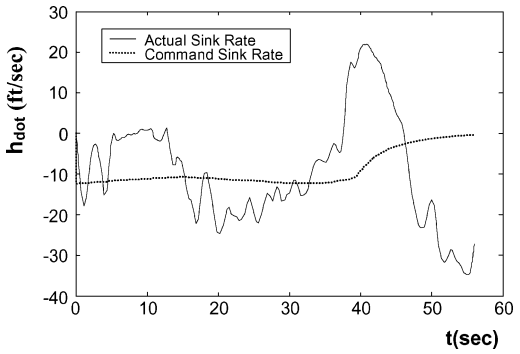

Fig. 36 Sink rate variations for neurocontroller with very strong wind.

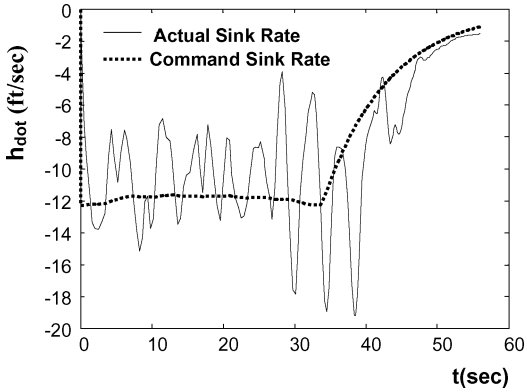

Fig. 37 Sink rate variations for neuro–PID controller with very strong wind.

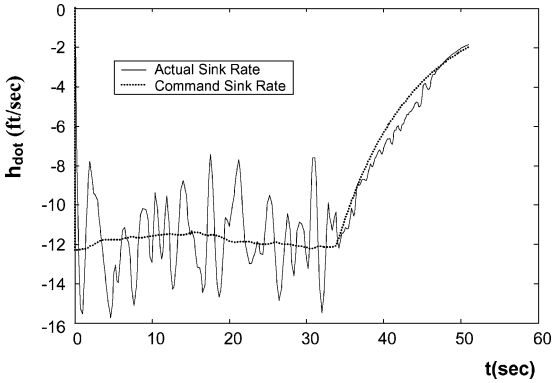

Fig. 38 Sink rate variations for ANFIS–PID controller with very strong wind.

controllers, and so the number of required sensors for all controllers is the same.

When the second aspect is considered, the neuro–PID controller due to the GRNN network, does require a great amount of calculations compare to the other structures. This could be a drawback only in cases where the time needed to generate a trajectory is comparable to the overall time for the mission. Fortunately, this does not hold for landing flight phase. Moreover, calculations could be done using the new generation of high-speed computers.

From the third aspect, only neuro- and adaptive controllers do not need any switching. This switching is associated with some practical

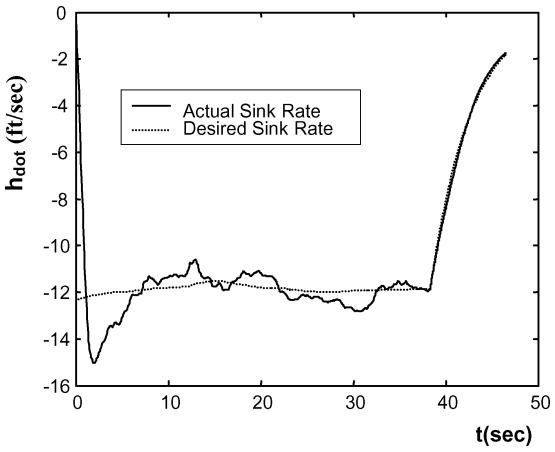

Fig. 39 Sink rate variations for adaptive controller with very strong wind.

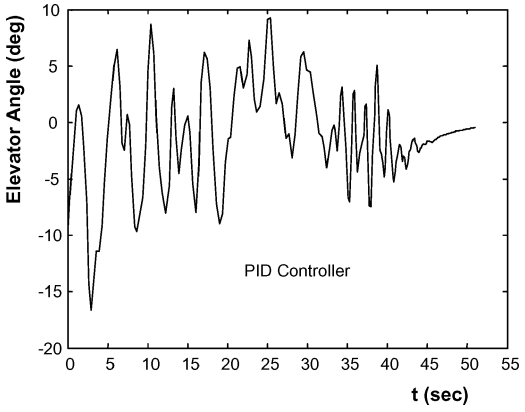

Fig. 40 Elevator control variations for PID controller with very strong wind.

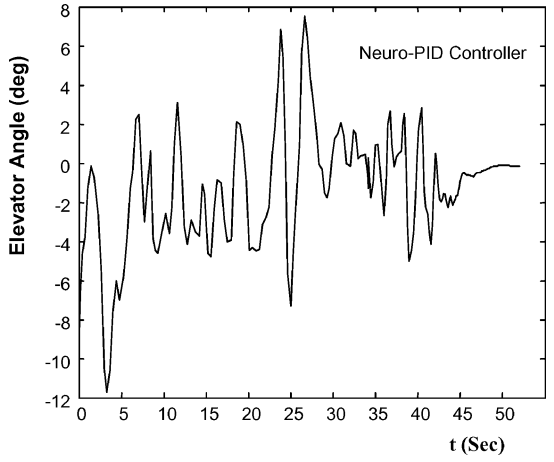

Fig. 41 Elevator control variations for hybrid nero–PID controller with very strong wind.

problems. For example, the process requires the exact information of sensors to switch between the glide and flare modes in the vicinity of the ground, which is hard to achieve without any error. Switching also introduces some noise in the electronics of the controller.

Finally, notice the number of sharp commands to the elevator from the adaptive controller, which might not be practical for existing technology (Figs. 40–43). For other controllers, the fluctuations are reasonable and do not pose any problem for actuators. Extensive simulation conducted by the authors show that through decreasing γ in Eq. (28) this problem almost disappears; however, the desired performance will be lost. On the other hand, for the ANFIS–PID controller in Fig. 42, it is relatively clear that the number of fluctuations is lower than that of the adaptive controller, once again, showing the better performance of ANFIS–PID controller.

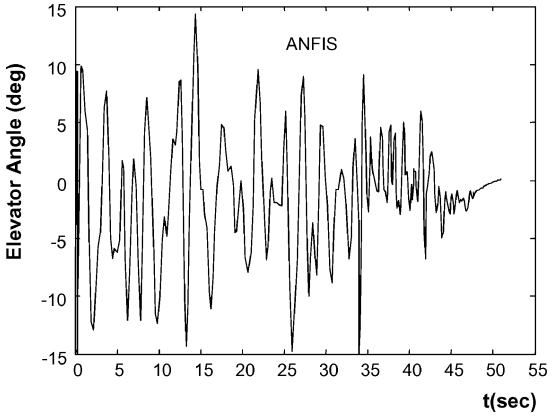

Fig. 42 Elevator control variations for ANFIS–PID controller with very strong wind.

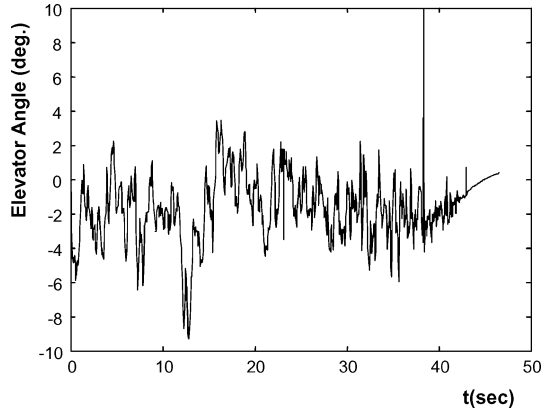

Fig. 43 Elevator control variations for adaptive controller with very strong wind.

VI. Conclusions

To expand the flight safety envelop of a typical transport jet in the landing phase of flight, five different controllers including a classical PID controller, two neural networks-based controllers, a fuzzy-based controller, and an adaptive controller have been evaluated in the presence of different wind shears and very strong wind patterns. Simulation results show that both fuzzy-based and adaptive controllers meet the necessary performance requirements and have acceptable robustness; however, the adaptive controller due to its large number of fluctuations in control signal is hard to implement. The fuzzy-based controller satisfies all necessary conditions for the selected performance specification and is a good candidate for expanding the flight envelop of aircraft in the landing phase of flight.

It is needless to stress that any new type of controller can only be judged comprehensively when it is actually built and tested in a real environment. There are always complex phenomena, such as noise, time delays, and quantization, that can severely impact controller performance and capabilities that must be considered before a final judgment is made to employ a system in any type of aircraft. However, the concept of a learning capable control system stems from human being functional architecture and deserves considerable research and attention.

References

[1]Pourtakdost, S. H., and Fazelzadeh Haghighi, S. A., "Guidance and Navigation During Landing and Optimal-Based Autolanding System Design," *Steghlal Journal*, Vol. 1, Sept. 1999, pp. 113–125 (in Persian).

[2]Jeppesen Sanderson, Inc., *Private Pilot Manual*, Englewood, CO, 1992.

[3]Mclean, D., *Automatic Flight Control Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1990.

[4]Shen, J., Park, E. K., and Bach, R. E., "Comprehensive Analysis of Two Downburst-Related Aircraft Accidents," *Journal of Aircraft*, Vol. 33, No. 5, 1996, pp. 924–930.

[5]Iiguni, Y., Akiyoshi, H., and Adachi, N., "An Intelligent Landing System Based on a Human Skill Model," *Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 3, 1998, pp. 877–882.

[6]Shue, S. P., and Agrawal, R. K., "Design of Automatic Landing Using Mixed H_2/H_∞ Control," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 1, 1999, pp. 103–114.

[7]Kaminer, I., and Khargonekar, P. P., "Design of the Flare Control Law for Longitudinal Autopilot Using H_∞ Synthesis," *Proceedings of the 29th Conference on Decision and Control*, IEEE, Dec. 1990, pp. 2981–2985.

[8]Ghalia, M. B., and Alouani, A. T., "Robust Control Design of an Autolanding System," *Proceedings of the 25th IEEE Southeastern Symposium on System Theory*, March 1993, pp. 248–252.

[9]Saini, G., and Balakrishnan, S. N., "Adaptive Critic Based Neuro-Controller for Autolanding of Aircraft," *Proceedings of the American Control Conference*, Vol. 2, June 1997, pp. 1081–1085.

[10]Izadi, H. A., Pakmehr, M., and Sadati, N., "Optimal Neuro-Controller in Longitudinal Autolanding of a Commercial Jet Transport," *Proceedings of the 2003 IEEE Conference on Control Applications (CCA03)*, Vol. 1, 2003, pp. 492–497.

[11]Juang, J. G., Chang, H. H., and Cheng, K. C., "Intelligent Landing Control Using Linearized Inverse Aircraft Model," *Proceedings of the 2002 American Control Conference*, Vol. 4, 2002, pp. 3269–3274.

[12]Malaek, S. M. B., Izadi, H. A., and Pakmehr, M., "Autolanding Controller Design Using Adaptive Control," *Advances in Dynamics, Instrumentation and Control*, 1st ed., edited by C. Y. Su, S. Rakheja, E. Wang, and R. B. Bhat, World Scientific, Nanjing, PRC, 2004, pp. 188–199.

[13]Moghadam, M. M., Izadi, H. A., and Pakmehr, M., "Neuro-Based Longitudinal Autolanding Controller Design of a Commercial Transport," *Proceedings of the 4th International Aerospace Conference* (Aero2003), Vol. 4, 2003, pp. 321–329.

[14]Malaek, S. M. B., Izadi, H. A., and Pakmehr, M., "Intelligent Autolanding Controller Based on Neural Networks," *Proceedings of the IFAC 1st African Control Conference (AFCON2003)*, Dec. 2003.

[15]Malaek, S. M. B., Sadati, N., Izadi, H. A., and Pakmehr, M., "Intelligent Autolanding Controller Based on Neural Networks and Fuzzy Logic," *Proceeding of 5th Asian Conference on Control Engineering*, July 2004, pp. 365–373.

[16]Roskam, J., *Airplane Flight Dynamics and Automatic Flight Control–Part II*, DARcorporation, Lawrence, KS, 1995.

[17]Hueschen, R. M., "Implementation and Flight Tests for the Digital Integrated Automatic Landing System (DIALS), Part 2," NASA-TM-87632-PT-2, July 1986.

[18]Heffley, R. K., Schulman, T. M., and Clement, W. F., "An Analysis of Airline Landing Flare Data Based on Flight and Training Simulator Measurements," NASA-CR-166404, Aug. 1982.

[19]Zheng, F., Wang, Q. G., and Lee, T. H., "Brief Paper on the Design of Multivariable PID Controllers via LMI Approach," *Journal of Automatica*, Vol. 38, No. 3, 2002, pp. 517–526.

[20]Martens, D., "Neural Networks as a Tool for the Assessment of Human Pilot Behavior in Wind Shear," *Aerospace Science and Technology Journal*, Vol. 3, No. 1, 1999, pp. 39–48.

[21]Demuth, H., and Beale, M., "Neural Networks Toolbox (For Use with MATLAB) User's Guide," Ver. 4, Mathworks, Inc., Natick, MA, 2000.

[22]Shine, J., and Jang, R., "ANFIS: Adaptive Network-Based Fuzzy Inference System," *IEEE Transactions on Systems*, Vol. 23, No. 3, 1993, pp. 665–685.

[23]"Fuzzy Logic Toolbox (For Use with MATLAB) User's Guide," Ver. 2, Mathworks, Inc., Natick, MA, 2002.

[24]Astrom, K. J., and Wittenmark, B., *Adaptive Control*, 2nd ed., Addison-Wesley, New York, 1995.

[25]"Simulink Toolbox of MATLAB (Model-Based and System-Based Design) User's Guide," Ver. 5, Mathworks, Inc., Natick, MA, 2002.

[26]"Automatic Landing Systems," Federal Aviation Administration, FAA Rept. AC 20-57A, Jan. 1971.