Journal of Universal Computer Science, vol. 15, no. 13 (2009), 2586-2607
submitted: 31/10/08, accepted: 13/6/09, appeared: 1/7/09 © J.UCS

Applications of Cerebellar Model Articulation Controllers to Intelligent Landing System

Jih-Gau Juang
(National Taiwan Ocean University, Keelung, Taiwan
jgjuang@mail.ntou.edu.tw)

Chia-Lin Lee
(National Taiwan Ocean University, Keelung, Taiwan
m95670005@mail.ntou.edu.tw)

Abstract: The atmospheric disturbances affect not only flying qualities of an airplane but also flight safety. According to flight records, most aircraft accidents occurred during final approach or landing. If the flight conditions are beyond the preset envelope, the automatic landing system (ALS) is disabled and the pilot takes over. An inexperienced pilot may not be able to guide the aircraft to a safe landing at the airport when wind disturbance is encountered. This study proposes different cerebellar model articulation controllers (CMAC) to improve the performance of conventional ALS. A CMAC with general basis function (CMAC-GBF) and a type-2 fuzzy CMAC (FCMAC) are applied to construct intelligent landing system which can guide the aircraft to a safe landing in severe wind turbulence environment.

Keywords: Intelligent Landing System, CMAC, Fuzzy System, Turbulence, PID control
Categories: L.3.6

1 Introduction

On March 1, 2008, at Hamburg airport, a Lufthansa Airbus A320 tried to land in crosswind conditions which exceeded the limit for the aircraft and made the left wing touch ground. The pilots then performed a go around and successfully saved the aircraft from crashing. According to a survey of the National Transportation Safety Board (NTSB) [NASDAC, 2000], 22.6 percent of aircraft accidents in the years of 1989 to 1999 were weather related. Most aircraft accidents occurred in final approach or landing. Another NTSB report, between 1994 and 2003, there were 19562 aircraft accidents. Weather was a contributing factor in 4159 of these accidents and involved 4167 aircraft. Of the 4159 weather-related accidents, 2726 were due to wind conditions. In addition, a single accident may involve multiple weather conditions. According to the statistics of Flight International 10-16, January 2006 issue [FSF, 2006], there were 23 accidents/incidents affected by weather, causing total 324 (34 crew and 290 passengers) fatalities. The average accident fatality caused by weather is 14 people. It was apparent that most of cases were in the landing phase. Therefore, pilots should never be absent-minded and remain their best condition, especially in landing phase.

The first Automatic Landing System (ALS) was made in England in 1965. Since

then, most aircraft have had this system installed. The ALS relies on the Instrument Landing System (ILS) to guide the aircraft into the proper altitude, position, and approach angle during the landing phase. According to the Federal Aviation Administration (FAA) regulations [FAA, 1997], environmental conditions considered the determination of dispersion limits as being: headwinds up to 25 knots, tailwinds up to 10 knots, crosswinds up to 15 knots, moderate turbulence, and wind shear of 8 knots per 100 feet from 200 feet to touchdown. If the flight conditions are beyond the preset envelope, the ALS is disabled and the pilot takes over. An inexperienced pilot may not be able to guide the aircraft to a safe landing at the airport. It is therefore desirable to develop an intelligent ALS that expands the operational envelope to include safer responses under a wider range of conditions. The goal of this paper is to show that the proposed intelligent ALS can relieve human operators and guide the aircraft to a safe landing in a severe turbulence environment.

In past decades, most of the improvements in the ALS system have been on the guidance instruments, such as GNSS Integrity Beacons, Global Positioning System, Microwave Landing System, and Automatic Land Position Sensor [Cohen, 1995] [DDC-I, 1995] [Asai, 1997] [Kaufmann, 1995]. By using improvement calculation methods and high accuracy instruments, these systems provide more accurate flight data to the ALS to make the landing smoother. However, these researches do not include weather factors such as wind turbulences. Recently, intelligent concepts such as neural networks, fuzzy system, genetic algorithm, and hybrid systems have applied to flight control to increase the flight controller's adaptively to different environments [Jorgensen, 1997] [Juang, 2004] [Chaturvedi, 2002] [Izadi, 2003] [Iiguni, 1998]. This paper proposes an intelligent aircraft automatic landing system that uses CMAC-GBF [Chiang, 2007] [Chang, 2003] [Chiang, 2005] [Chuang, 2007] and type-2 FCMAC [Liu, 2007] [Wang, 2004] [Liang, 2000] to improve the performance of conventional ALS. Comparisons of conventional CMAC [Albus, 1975] [Albus, 1975] and conventional (type-1) FCMAC [Juang, 2008] are also given. The performance of the intelligent ALS under severe environment can be improved by the advantages of the CMAC which include local generalization and rapid learning process.

2 Landing System

At the aircraft landing phase, the pilot descends from the cruise altitude to an altitude of approximately 1200 feet above the ground. The pilot then positions the aircraft so that the aircraft is on a heading towards the runway centerline. When the aircraft approaches the outer airport marker, which is about 4 nautical miles from the runway, the glide path signal is intercepted, as shown in [Fig. 1]. As the airplane descends along the glide path, its pitch, attitude, and speed must be controlled. The descent rate is about 10 ft/sec and the pitch angle is between -5 to +5 degrees. Finally, as the airplane descends 20 to 70 feet above the ground, the glide path control system is disengaged and a flare maneuver is executed. The vertical descent rate is decreased to 2ft/sec so that the landing gear may be able to dissipate the energy of the impact at landing. The pitch angle of the airplane is then adjusted, between 0 to 5 degrees for most aircraft, which allows a soft touchdown on the runway surface.

A simplified model of a commercial aircraft that moves only in the longitudinal and vertical plane is used in the simulations for implementation ease [Jorgensen, 1997]. The motion equations of the aircraft are given as follows:

$$\dot{u} = X_u(u-u_g) + X_w(w-w_g) + X_q\Delta q - g(\frac{\pi}{180})\cos(\gamma_0)\Delta\theta + Z_E\delta_E + Z_T\delta_T \quad (1)$$

$$\dot{w} = Z_u(u-u_g) + Z_w(w-w_g) + (Z_q - \frac{\pi}{180}U_0)\Delta q - g(\frac{\pi}{180})\sin(\gamma_0)\theta_T \quad (2)$$

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

$$\dot{\theta} = q \quad (4)$$

$$\dot{h} = -w + \frac{\pi}{180}U_0\theta \quad (5)$$

where u is the aircraft longitudinal velocity (ft/sec), w is the aircraft vertical velocity (ft/sec), q is the pitch rate (rate/sec), θ is the pitch angle (deg), h is the aircraft altitude (ft), δ_E is the incremental elevator angle (deg), δ_T is the throttle setting (ft/sec), γ_o is the flight path angle (-3deg), and g is the gravity (32.2 ft/sec^2). The parameters X_i, Z_i and M_i are the stability and control derivatives.

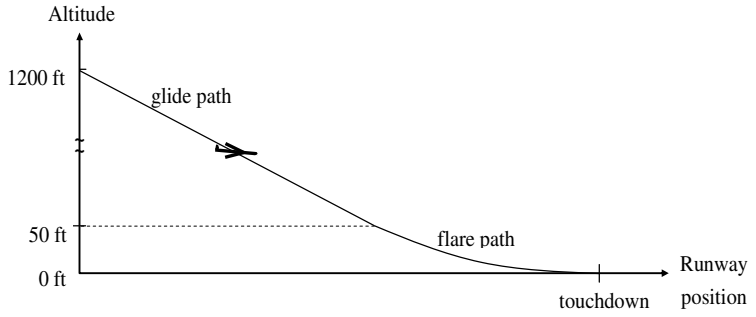

Figure 1: Glide path and flare path

To make the ALS more intelligent, reliable wind profiles are necessary. Two spectral turbulence forms models by von Karman and Dryden are mostly used for aircraft response studies. In this study the Dryden form [Jorgensen, 1997] was used for its demonstration ease. The model is given by :

$$u_g = u_{gc} + N(0,1)\sqrt{\frac{1}{\Delta t}}\left(\frac{\sigma_u\sqrt{2a_u}}{s+a_u}\right) \quad (6)$$

$$w_g = N(0,1)\sqrt{\frac{1}{\Delta t}}\left(\frac{\sigma_w\sqrt{3a_w}(s+b_w)}{(s+a_u)^2}\right) \quad (7)$$

where $u_{gc} = -u_{wind\,510}[1+\frac{\ln(h/510)}{\ln(51)}]$, $a_u = \frac{U_o}{L_u}$, $L_w = h$, $a_u = \frac{U_o}{L_u}$, $a_w = \frac{U_o}{L_w}$,

$b_w = \frac{U_o}{L_w\sqrt{3}}$, $L_u = 100h^{1/3}$ for $h > 230$, $L_u = 600$ for $h \le 230$,

$\sigma_w = 0.2\left|u_{gc}\right|(0.5 + 0.00098 \times h)$ for $0 \le h \le 500$, $\sigma_w = 0.2\left|u_{gc}\right|$ for $h > 500$.

The parameters are: u_g is the horizontal wind velocity (ft/sec), w_g is the vertical wind velocity (ft/sec), U_0 is the nominal aircraft speed (ft/sec), $u_{wind\,510}$ is the wind speed at 510 ft altitude, L_u and L_w are scale lengths (ft), σ_u and σ_w are RMS values of turbulence velocity (ft/sec), Δt is the simulation time step (sec), *N(0,1)* is the Gaussian white noise with zero mean and unity standards deviation, u_{gc} is the constant component of u_g, and h is the aircraft altitude (ft). [Fig. 2] shows a turbulence profile with a wind speed of 30 ft/sec at 510 ft altitude.

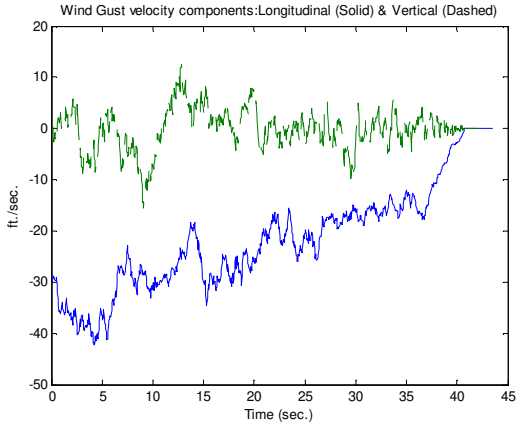

Figure 2: Turbulence profile

3 Control Scheme

Conventional aircraft landing system uses PID-type control, as shown in [Fig. 3]. Controller inputs consist of altitude and altitude rate commands along with aircraft altitude and altitude rate. The pitch command θ_c is obtained from the PID controller. Then, the pitch autopilot is controlled by pitch command. The pitch autopilot is shown in [Fig. 4]. In order to enable aircraft to land more steady when an aircraft arrives to the flare path, a constant pitch angle will be added to the controller. In general, the PID controller is simple and effective but there are some drawbacks such as apparent overshoot and sensitive to external noise and disturbance. When severe turbulence is encountered the PID controller may not be able to guide the aircraft to land safely. With CMAC compensator the proposed controller can overcome these disadvantages. It uses a traditional PID controller to stabilize the system and train the CMAC to provide precise control. The gains of PID controller are adjusted based on

experiences, what it provides are tolerable solutions, not desired solutions. The CMAC can effectively meliorate these conditions.

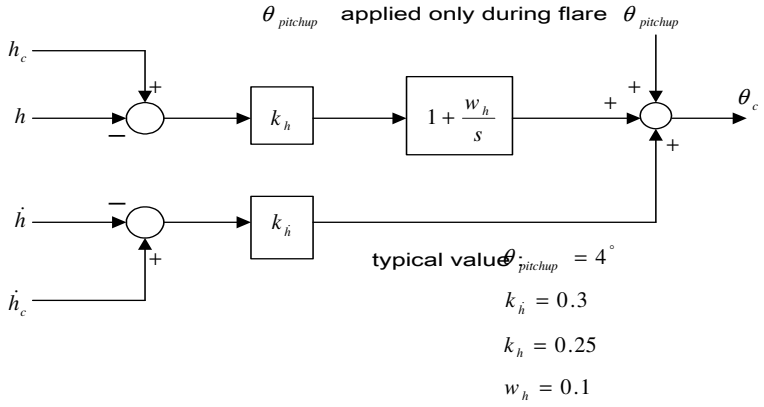

Figure 3: PID-controller

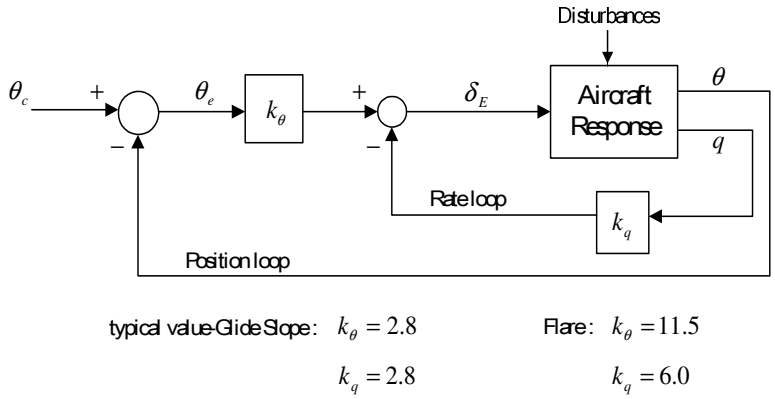

Figure 4: Pitch autopilot

The overall control scheme is described in [Fig. 5], in which the control signal U is the sum of the PID controller output and the CMAC-GBF or type-2 fuzzy CMAC output. The inputs for the CMAC and PID controller are: altitude, altitude command, altitude rate, and altitude rate command. The PID controller provides tolerable solutions. In each time step k, the CMAC involves a recall process and a learning process. In the recall process, it uses the desired system output of the next time step and the actual system output as the address to generate the control signal U_{CMAC}. In the learning process, the control signal of the pitch autopilot, U, is treated as a desired output. It is used to modify the weights of CMAC stored at location which is addressed by the actual system output and the system output of the next time step. The output of the CMAC is the compensation for pitch command. When the wind turbulence is too strong, the ALS can not control the aircraft to land safely. Here we

use a CMAC-GBF or a type-2 fuzzy CMAC control scheme to improve the ability of turbulence resistance of the ALS.

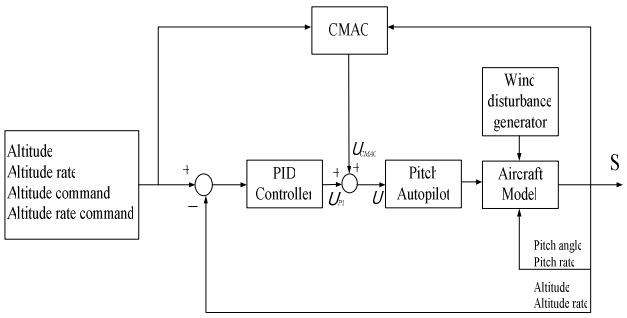

Figure 5: The CMAC control scheme

3.1 Cerebellar Model Articulation Controller (CMAC)

CMAC is a type of artificial neural network proposed in the literatures [Albus, 1975]. It could be considered as an associative memory learning structure based on the performance of the cerebellum of human being. The function of CMAC is alike to a lookup-table technique which represents complex and nonlinear systems. And the fundamental concept of CMAC is to store information into overlapping regions in an associative approach so that stored information can easily be recalled using less storage space (memory cell). The structure of CMAC is shown in [Fig. 6]. Manipulation of the CMAC divides the algorithm into two segments.

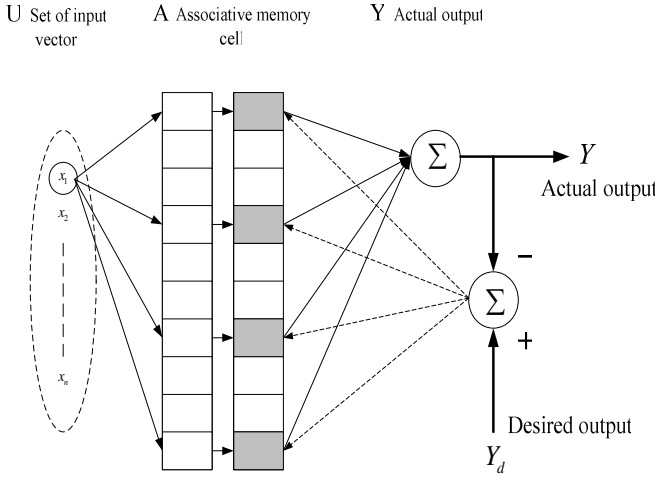

Figure 6: The conceptual diagram of CMAC

First is the output generating stage. The output of CMAC can be obtained by the mapping process U → A → Y, where A stands for the *M*-dimensional memory cell, the $a \in A \subset R_M$ is the binary associative vector, as an address indexes in coherence

with the input vector x. Let the input x address N ($N < M$) memory cells; the mapping A → Y represents the chosen weights that stored in memory cells are added together to compute the output as:

$$y(x) = \sum_{j=1}^{N} w_j a_j(x) \tag{8}$$

where w_j is the weight of the j^{th} storage hypercube and $a_j(x)$ is a binary factor indicating whether the j^{th} storage hypercube is addressed by the input x. Second is the stage of network learning in the CMAC, it is to update the addressed weights of memory cells according to the error between the desired output and the real output. Its weight updating rule is:

$$w_j^{(i)} = w_j^{(i-1)} + \frac{\alpha}{m}(y_d - \sum_{j=1}^{N} w_j^{(i-1)} a_j) \tag{9}$$

where y_d is the desired output, m is the number of addressed memory cells, α is the learning rate.

When it processes input vector of the CMAC, it simply divides it into certain blocks. The relation between input vector with these blocks is simply a crisp relation. The relation between the input condition and the association intensity is simply "activated" or "not activated". Further, an important identity of the CMAC is local generalization that derived from where nearby input vectors have some overlapping vicinity and then share some associative memory cells.

3.2 CMAC-GBF

The CMAC represents one kind of associative memory technique. In the addressing technique, each input space (state variable) is quantized and the output space is divided into discrete states. A quantized input vector specifies a discrete state and is used to generate addresses for retrieving information from memory for this state. There is no big difference on the structure and quantization method and the mapping process between CMAC-GBF and original CMAC. The difference is that while the input vector maps the hypercube by way of the association in [Fig. 7], the stored data can be the differential basis function value.

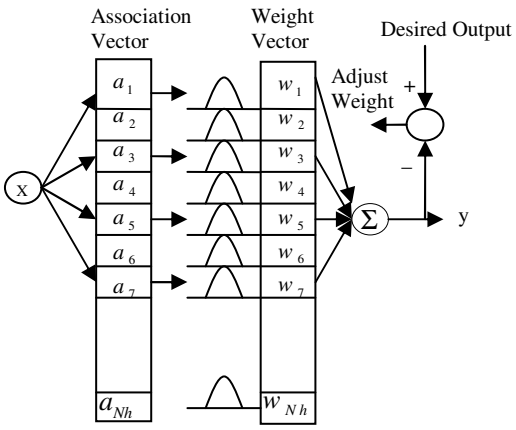

Figure 7: The structure of CMAC-GBF

[Fig. 8] illustrates the block division of the CMAC-GBF for a two-variable case. This simple example has two state variables (x_1 and x_2) with each quantized into four discrete regions, called blocks. For instance, x_1 can be divided into *A*, *B*, *C,* and *D* and x_2 can be divided into *a*, *b*, *c,* and *d*. Areas *Aa*, *Ab*,...*Dd* formed by quantized regions are called hypercubes. By shifting each block a small interval (called an element), different hypercubes can be obtained. For instance, *E*, *F*, *G*, and *H* in the second row for x_1 and *e*, *f*, *g*, and *h* in the second row for x_2 are possible shifted regions. *Ee*, *Ef*,...*Hh* are new hypercubes from the shifted regions. With this kind of decomposition, if there are *Ne* elements in a complete block, we will have *Ne* layers of hypercubes. In the given example, there are four layers as shown in [Fig. 8]. The state is covered by *Ne* different hypercubes, one from each layer.

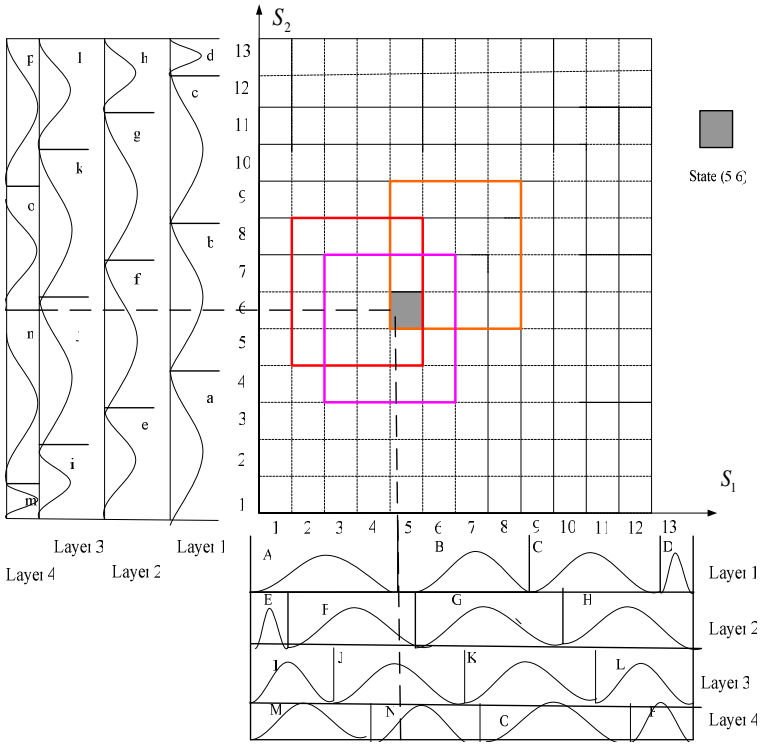

Figure 8: Block division of CMAC-GBF for a two-variable case

In the original CMAC, a constant value is assigned to each hypercube. By inputting vectors into these hypercubes, no matter what the positions are, the obtained values are the same. The output of CMAC is the sum of several values of hypercube and the input vector is treated as memory addresses. Therefore, the output is irrelevant to input and the differential of output to input is also not obtainable. Thus, in the CMAC-GBF, a general basis function (GBF) is used to replace constants in every hypercube. In this way, the positions of input vectors are related to output value, and the learning precision is also increased.

In the CMAC-GBF, the content of hypercube can be expressed as $w_i(x_s) = v_i b_i(x_s)$, where $b_i(x_s)$ is a general basis function and v_i is a weight to be obtained through learning. The output of the CMAC-GBF can be written as below.

$$y(x_s) = a_s^T w(x_s) = \sum_i^{N_e} [a_{s,i} . v_i . b_i(x_s)] \tag{10}$$

where $a_s^T = [a_{s,1}\ a_{s,2} .....a_{s,N_e}]$ is the hypercube selection vector and $w = [w_1\ w_2 .....w_{N_e}]$ is the vector of memory contents of the CMAC-GBF. The Gaussian functions are employed as the basis functions

$$b_i(x_s) \equiv \prod_{j=1}^{N_v} \phi_{ij}(x_{s,i}) \tag{11}$$

with

$$\phi_{ij}(x_{s,i}) = \exp[-\frac{(x_{s,i} - m_{ij})^2}{\sigma_{ij}^2}] \tag{12}$$

where m_{ij} is the mean, σ_{ij} is the variance and N_v is the number of variables in the target function. Consequently, the weight function is

$$w_i(x_s) = v_i \prod_{j=1}^{N_v} \phi_{ij}(x_{s,i}) \tag{13}$$

The output from the CMAC with Gaussian basis functions can be mathematically expressed as

$$E = \frac{1}{2}(\hat{y}_s - y_s)^2 \tag{14}$$

the updated amount for v_i can be set equal to

$$\begin{aligned} \Delta v_i &= -\frac{\alpha_v}{N_e}\frac{\partial E}{\partial v_i} = -\frac{\alpha_v}{N_e}\frac{\partial E}{\partial w_i(x_s)}\frac{\partial w_i(x_s)}{\partial v_i} \\ &= -\frac{\alpha_v}{N_e}(\hat{y}_s - a_s^T w(x_s))a_{s,i}b_i(x_s) \end{aligned} \tag{15}$$

where α_v is the learning rate for *v*.

The means and variances of the Gaussian functions can also be adjusted to increase the approximation capability. The updating rules for these parameters can be derived as

$$\begin{aligned} \Delta m_{ij} &= -\frac{\alpha_m}{N_e}\frac{\partial E}{\partial m_{ij}} = -\frac{\alpha_m}{N_e}\frac{\partial E}{\partial w_i(x_s)}\frac{\partial w_i(x_s)}{\partial \phi_{ij}}\frac{\partial \phi_{ij}}{\partial m_{ij}} \\ &= -\frac{\alpha_m}{N_e}(\hat{y}_s - a_s^T w(x_s))a_{s,i}v_i b_i(x_s)\frac{2(x_{s,i} - m_{ij})}{\sigma_{ij}^2} \end{aligned} \tag{16}$$

$$\begin{aligned} \Delta \sigma_{ij} &= -\frac{\alpha_\sigma}{N_e}\frac{\partial E}{\partial \sigma_{ij}} \\ &= -\frac{\alpha_\sigma}{N_e}\frac{\partial E}{\partial w_i(x_s)}\frac{\partial w_i(x_s)}{\partial \phi_{ij}}\frac{\partial \phi_{ij}}{\partial \sigma_{ij}} \\ &= -\frac{\alpha_m}{N_e}(\hat{y}_s - a_s^T w(x_s))a_{s,i}v_i b_i(x_s)(\frac{2(x_{s,i} - m_{ij})^2}{\sigma_{ij}^3}) \end{aligned} \tag{17}$$

where α_m and α_σ are the learning rates for the variances.

3.3 Type-2 Fuzzy CMAC

The type-2 fuzzy theorem is utilized into CMAC structure in order to promote more accurate resolution than conventional CMAC. The mapping procedure of type-2 FCMAC is similar to conventional FCMAC. The diagram structure of type-2 FCMAC is shown in [Fig. 9]. Each phase of mapping is described as follows. The X is an n-dimensional input space, as shown in [Fig. 10]. For the given $X = [x_1, x_2 ..... x_n]$, $S = [s_1, s_2 ..... s_n]$ represents the quantization vector of x. It is specified the corresponding state of each input variable before the fuzzification. Type-2 FCMAC uses the interval type-2 fuzzification method of the fuzzy theorem as its addressing scheme. After the input vector to the interval type-2 fuzzy set is being fuzzified, the input state values are transformed to upper firing strength and lower firing strength, which is based on corresponding interval type-2 membership functions. We choose the product inference method as the t-norm operator. The j^{th} rule's upper firing strength $\overline{c}^{\,j}$ and lower firing strength firing strength $\underline{c}^{\,j}$ in type-2 FCMAC could be computed as:

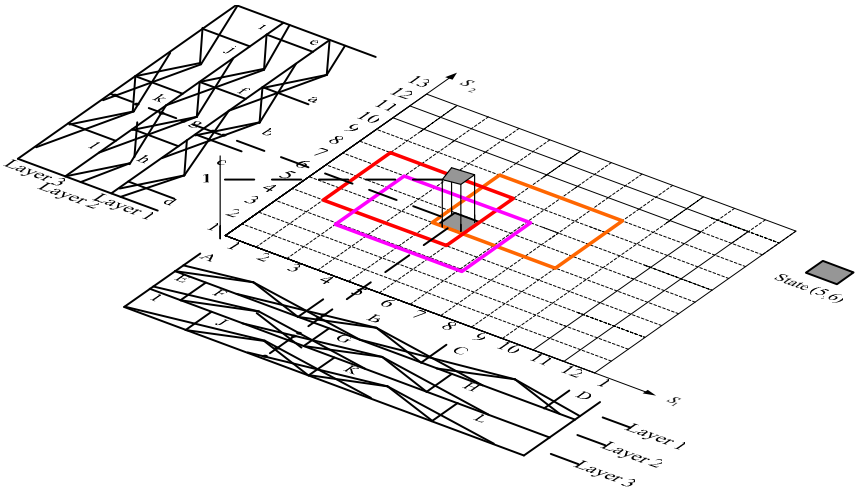

Figure 9: Diagram of type-2 FCMAC in 3-D

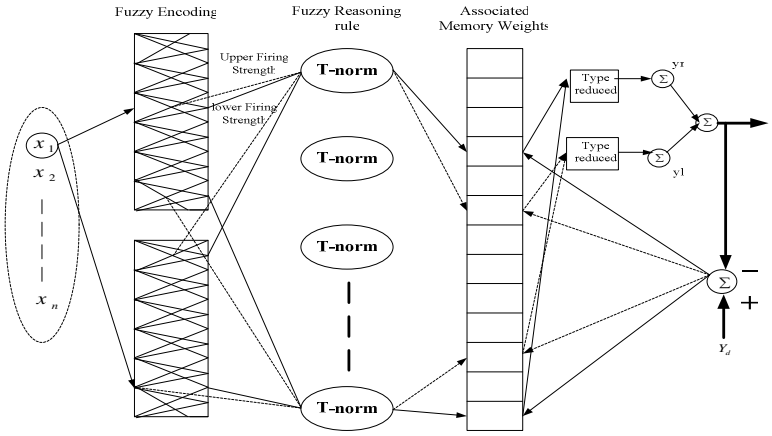

Figure 10: Architecture of type-2 FCMAC network

$$\bar{c}^{\,j}(x) = \bar{c}_{j_1}(x_1) * \bar{c}_{j_2}(x_2) * .....\bar{c}_{j_n}(x_n) = \prod_{i=1}^{n} \bar{c}_{j_i}(x_i) \tag{18}$$

$$\underline{c}^{\,j}(x) = \underline{c}_{j_1}(x_1) * \underline{c}_{j_2}(x_2) * .....\underline{c}_{j_n}(x_n) = \prod_{i=1}^{n} \underline{c}_{j_i}(x_i) \tag{19}$$

The type-reduced set of the Type-2 FCMAC using the center of sets type reduction :

$$y_{\cos} = [y_l, y_r] = \int_{w^1 \in [\underline{w}^1, \overline{w}^1]} .... \int_{w^N \in [\underline{w}^M, \overline{w}^N]} ... \int_{c^1 \in [\underline{c}^1, \overline{c}^1]} ..... \int_{c^M \in [\underline{c}^M, \overline{c}^N]} 1 \Big/ \frac{\sum_{j=1}^{n} c^j w^j}{\sum_{j=1}^{n} c^j} . \tag{20}$$

It is an interval type-1 set determined by its left and right end points y_l and y_r, which can be written as follows [Liang, 2000]:

$$y_r = \frac{\sum_{j=1}^{N} \bar{c}^{\,j} \overline{w}^{\,j}}{\sum_{j=1}^{N} \bar{c}^{\,j}} = \frac{\sum_{j=1}^{R} \underline{c}^{\,j} \overline{w}^{\,j} + \sum_{j=R+1}^{N} \bar{c}^{\,j} \overline{w}^{\,j}}{\sum_{j=1}^{R} \underline{c}^{\,j} + \sum_{j=R+1}^{N} \bar{c}^{\,j}} \tag{21}$$

$$y_l = \frac{\sum_{j=1}^{N} \underline{c}^{\,j} \underline{w}^{\,j}}{\sum_{j=1}^{N} \underline{c}^{\,j}} = \frac{\sum_{j=1}^{L} \bar{c}^{\,j} \underline{w}^{\,j} + \sum_{j=L+1}^{N} \underline{c}^{\,j} \underline{w}^{\,j}}{\sum_{j=1}^{L} \bar{c}^{\,j} + \sum_{j=L+1}^{N} \underline{c}^{\,j}} \tag{22}$$

$\overline{w}$ and $\underline{w}$ are the corresponding weights of $\overline{c}$ and $\underline{c}$, respectively. L and R can be obtained from [Liang, 2000]:

Step 1. Assume that the pre-computed $\overline{w}^j$ are arranged in ascending order, i.e., $\overline{w}^1 \le \overline{w}^2 \le ..... \le \overline{w}^N$

Step 2. Compute y_r by initially setting $\overline{c}^j = (\overline{c}^j + \underline{c}^j)/2$ for $j = 1.....N$ and let $y_r' = y_r$

Step 3. Find $R(1 \le R \le N-1)$ such that $\overline{w}^R \le y_r' \le \overline{w}^{R+1}$

Step 4. Compute y_r with $\overline{c}^j = \underline{c}^j$ for $j \le R$ and $\overline{c}^j = \overline{c}^j$ for $j > R$ and let $y_r'' = y_r$

Step 5. If $y_r'' \ne y_r$ then go to step 6. If $y_r'' = y_r'$ then stop and set $y_r'' \equiv y_r'$

Step 6. Set $y_r' = y_r''$ and return to Step 3.

The procedure for computing y_l is very similar to the one just given for y_r. In Step 3 find $L(1 \le L \le N-1)$ such that $\underline{w}^L \le y_l' \le \underline{w}^{L+1}$. Additionally, in Step 2 compute y_l initially setting $\underline{c}^j = (\overline{c}^j + \underline{c}^j)/2$ for $j = 1.....N$ and in Step 4 compute y_l with $\underline{c}^j = \overline{c}^j$ for $j \le L$ and $\underline{c}^j = \underline{c}^j$ for $j > L$.

The defuzzified output is simply the average

$$y = y_r + y_l \tag{23}$$

The work on learning of type-2 FCMAC is to update the memory weight according to the error between the desired output and the actual output. The learning rule for Type-2 FCMAC is as following:

$$\overline{w}_j^{(i)} = \overline{w}_j^{(i-1)} + \frac{\alpha}{m}(y_d - y)\overline{c}_j(x) / \sum_{j=1}^{N} \overline{c}_j(x) \tag{24}$$

$$\underline{w}_j^{(i)} = \underline{w}_j^{(i-1)} + \frac{\alpha}{m}(y_d - y)\underline{c}_j(x) / \sum_{j=1}^{N} \underline{c}_j(x) \tag{25}$$

where α is the learning rate, m is the size of floor (called generalization).

4 Simulations

The aircraft starts the initial states of the ALS as follows: the flight height is 500 ft, the horizontal position before touching the ground is 9240 ft, the flight angle is -3 degrees, the speed of the aircraft is 234.7 ft/sec. Successful touchdown landing conditions are defined as follows:

1). $-3 \le \dot{h}_{TD}(T) \le -1$ (ft/sec)

2). $-300 \le x_{TD}(T) \le 1000$ (ft)

3). $200 \le V_{TD}(T) \le 270$ (ft/sec)

4). $-10 \le \theta_{TD}(T) \le 5$ (degrees)

where T is the time at touchdown, $\dot{h}_{TD}$ is vertical speed of the aircraft at touchdown, x_{TD} is the horizontal position at touchdown, V_{TD} is the horizontal speed, θ_{TD} is the pitch angle at touchdown.

[Tab. 1] shows the results from using PID controller with different wind turbulence speeds. The conventional controller with original control gains can only successfully guide an aircraft flying through wind speeds of 0 ft/sec to 30 ft/sec. An aircraft is safe to land in the wind turbulence speed at 30 ft/sec. The situations at wind turbulence 30 ft/sec are that the pitch angle is -0.1664 degrees, vertical speed is -2.193 ft/sec, horizontal velocity is 234.6779 ft/sec, and horizontal position at touchdown is 843.9305 ft, as shown in Fig. 11 to Fig. 13. If the wind speed is higher than 30 ft/sec, the ALS will be unable to guide an aircraft to land safely. For the safe landing of an aircraft using the CMAC-GBF control scheme under different wind turbulence speeds are tested. The simulations show that the CMAC-GBF controller has well adaptive capability against severe wind disturbance. It is more robust than conventional PID type controller and original CMAC [Juang, 2008], as shown in [Tab. 2] and [Fig. 14] to [Fig. 16]. The CMAC-GBF control scheme can successfully guide the aircraft flying through wind speeds of 0 ft/sec to 71 ft/sec as shown in [Tab. 3]. The situations at wind turbulence 71 ft/sec are that the pitch angle is 0.8998 degrees, vertical speed is -1.7136 ft/sec, horizontal velocity is 234.6779 ft/sec, and horizontal position at touchdown is 820.4627 ft, as shown in [Fig.17] to [Fig. 19].

Wind speed	Landing Point (ft)	Aircraft vertical Speed (ft/sec)	Pitch angle (degree)
0	796	-2.83	-1.40
10	909	-2.54	-0.84
20	808	-2.37	-0.59
30	843	-2.19	-0.16
40	1019	-1.72	0.44

Table 1: Results from using conventional PID controller (k1=2.8; k2=2.8; k3=11.5; k4=6.0)

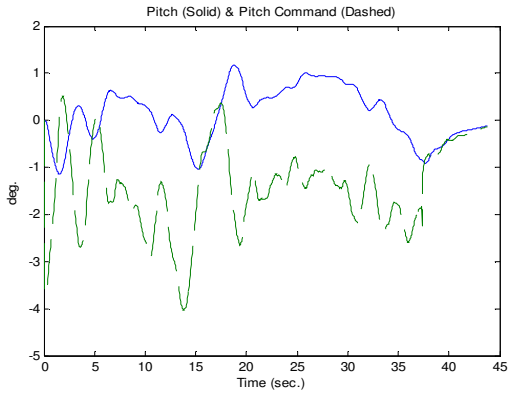

Figure 11: Aircraft pitch and pitch command using PID controller at turbulence 30 ft/sec

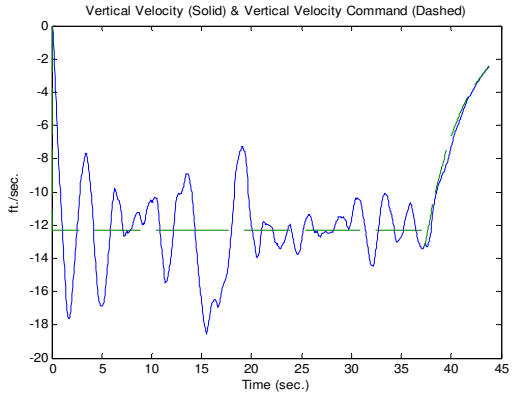

Figure 12: Aircraft vertical velocity and command using PID controller at turbulence 30 ft/sec

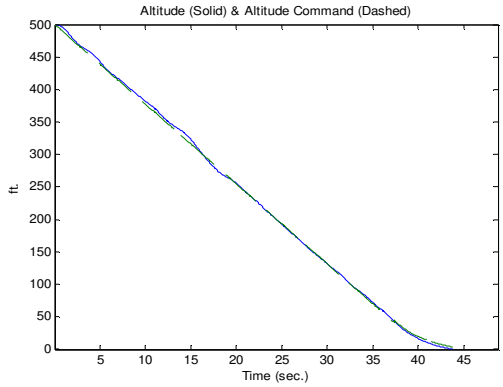

Figure 13: Aircraft altitude and command using PID controller at turbulence 30 ft/sec

Wind speed	Landing point (ft)	Aircraft vertical speed (ft/sec)	Pitch angle (degree)
0	854	-2.55	-0.96
10	762	-2.76	-0.93
20	774	-2.51	-0.61
30	844	-2.72	-0.41
40	691	-1.93	0.21
50	586	-2.26	0.87
58	844	-2.58	0.98

Table 2: Results from using CMAC controller

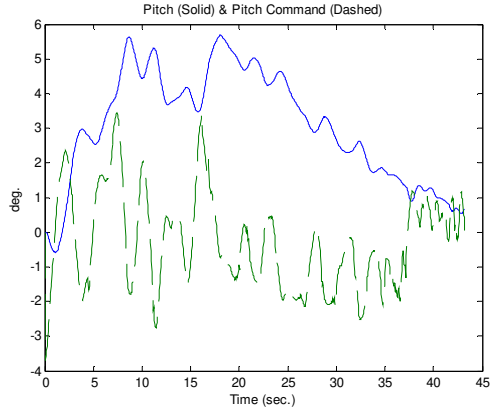

Figure 14: Aircraft pitch and pitch command using CMAC at turbulence 58 ft/sec

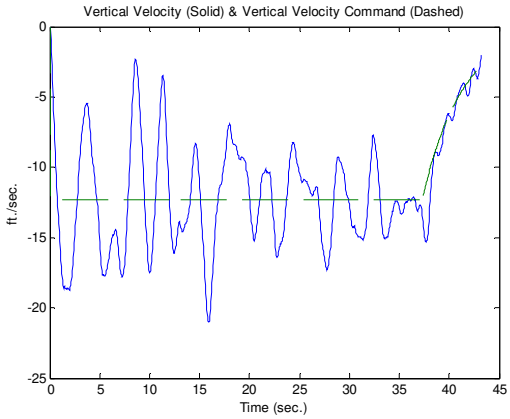

Figure 15: Vertical velocity and command using CMAC at turbulence 50 ft/sec

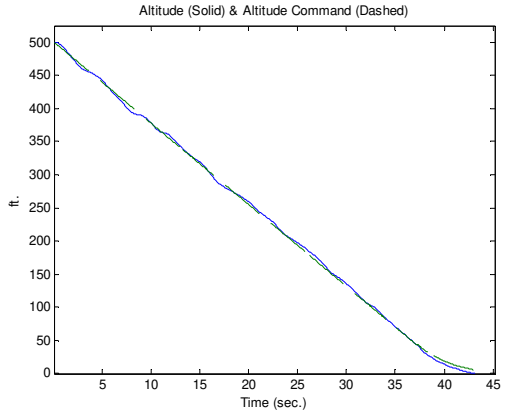

Figure 16: Aircraft altitude and command using CMAC at turbulence 50 ft/sec

Wind speed	Landing point (ft)	Aircraft vertical speed (ft/sec)	Pitch angle (degree)
10	796	-2.61	-0.95
20	808	-2.42	-0.54
30	949	-2.16	-0.33
40	796	-2.09	0.07
50	796	-2.13	0.17
60	808	-1.55	0.92
71	820	-1.71	0.89

Table 3: The results from using CMAC-GBF control scheme

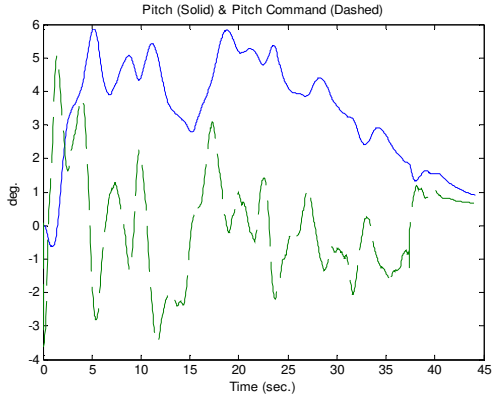

Figure 17: Aircraft pitch and pitch command using CMAC-GBF at turbulence 71 ft/sec

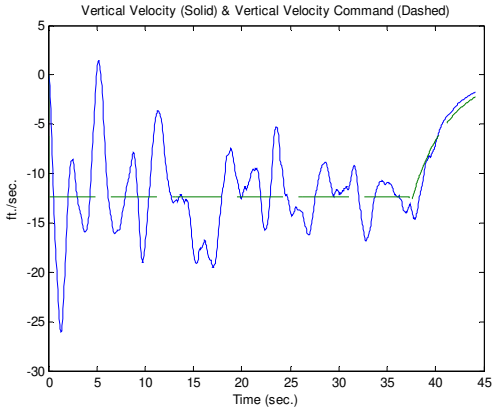

Figure 18: Aircraft vertical velocity and command using CMAC-GBF at turbulence 71 ft/sec

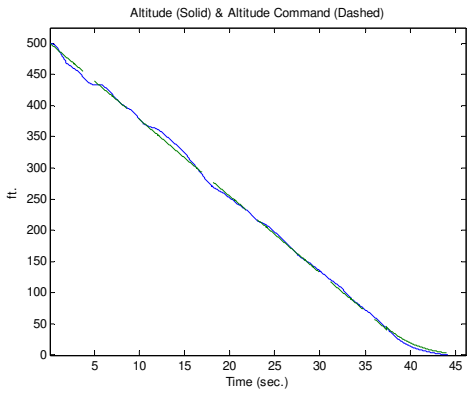

Figure 19: Aircraft altitude and command using CMAC-GBF at turbulence 71 ft/sec

The type-2 FCMAC control scheme can successfully guide the aircraft flying through wind speeds of 0 ft/sec to 100 ft/sec while the type-1 FCMAC can only reach 90 ft/sec [Juang, 2008], as shown in [Tab. 4] and [Fig. 20] to [Fig. 22]. [Tab. 5] shows the results from using type-2 FCMAC. The situations at wind turbulence 100 ft/sec are that the pitch angle is 0.6968 degrees, vertical speed is -1.7101 ft/sec, horizontal velocity is 234.6779 ft/sec, and horizontal position at touchdown is 937.8017 ft, as shown in [Fig. 23] to [Fig. 25].

Wind speed	Landing point (ft)	Aircraft vertical speed (ft/sec)	Pitch angle (degree)
10	797	-2.83	-1.41
30	938	-1.54	-0.58
50	891	-2.13	0.47
70	691	-2.21	1.41
90	926	-1.99	1.34

Table 4: Results from using type-1 FCMAC control

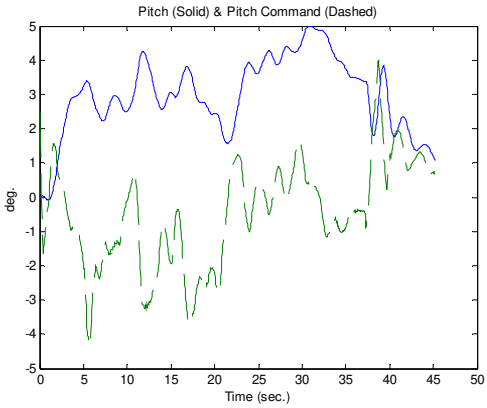

Figure 20: Pitch and pitch command using type-1 FCMAC at turbulence 90 ft/sec

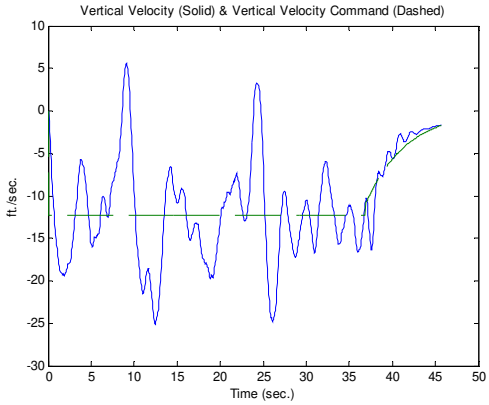

Figure 21: Aircraft vertical velocity and command using type-1 FCMAC at turbulence 90 ft/sec

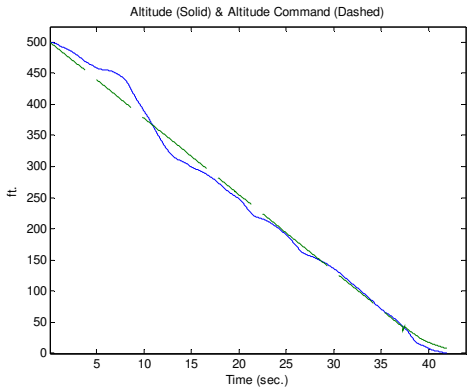

Figure 22: Aircraft altitude and command using type-1 FCMAC at turbulence 90 ft/sec

Wind speed	Landing point (ft)	Aircraft vertical speed (ft/sec)	Pitch angle (degree)
20	855	-2.51	-0.58
40	726	-2.44	0.03
60	996	-2.00	0.50
80	890	-1.71	1.39
100	937	-2.21	2.05

Table 5: The Results from using type-2 FCMAC control

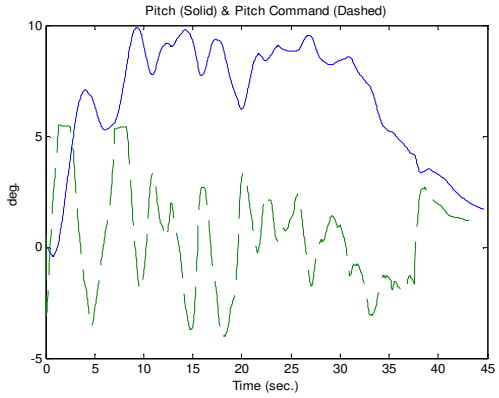

Figure 23: Pitch and pitch command form using type-2 FCMAC at turbulence 100 ft/sec

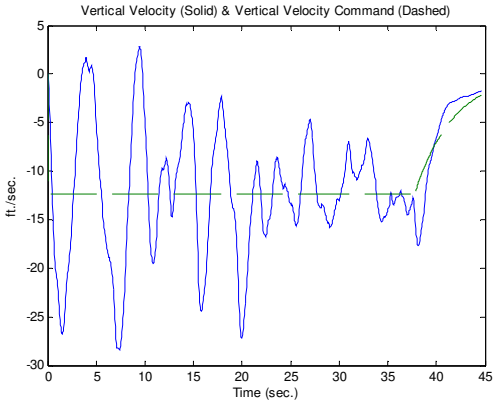

Figure 24: Vertical velocity and command form using type-2 FCMAC at turbulence 100 ft/sec

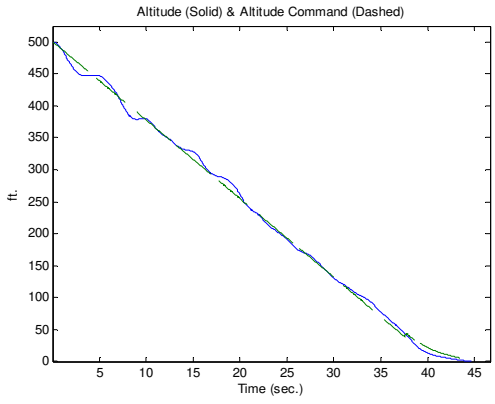

Figure 25: Altitude and command form using type-2 FCMAC at turbulence 100 ft/sec

5 Conclusions

The purpose of this paper is to investigate the use of a CMAC with general basis function and a type-2 fuzzy CMAC in aircraft automatic landing system and to make the automatic landing system more intelligent. Current flight control law is adopted in the intelligent controller design. Tracking performance and adaptive capability are demonstrated through software simulations. For the safe landing of an aircraft using a conventional controller, CMAC or a conventional fuzzy CMAC, the wind speed of turbulence limits are 30, 58, and 90 ft/sec, respectively. In this study, a well-trained CMAC-GBF control scheme can reach 71 ft/sec and the type-2 fuzzy CMAC can reach 100 ft/sec. The proposed controllers have better performance than previous works. The CMAC controller can be used to successfully replace the conventional controller. The proposed intelligent controller can act as an experienced pilot and guide the aircraft to a safe landing in severe wind turbulence environment.

Acknowledgement

This work was supported by the National Science Council, Taiwan, ROC under Grant NSC 92-2213-E-019 -005.

References

[Albus, 75] Albus, J.S.: "A New Approach to Manipulator Control: the Cerebellar Model Articulation Control (CMAC)," ASME Journal of Dynamic Systems, Measurement, and Control, vol.97, pp.220-227, Sep. 1975.

[Albus, 75] Albus, J.S.: Data Storage in the Cerebellar Model Articulation Controller (CMAC), ASME Journal of Dynamic Systems, Measurement, and Control, vol.97, pp.228-233, Sep. 1975.

[Asai, 97] Asai, S., et al.: Development of Flight Control System for Automatic Landing Flight Experiment, Mitsubishi Heavy Industries Technical Review, vol. 34, no. 3, 1997.

[Chang, 03] Chang, L.S., Lai, L.Y.: Differentiable Cerebellar Model Articulation Controller for Function Approximation and Motor Control, master thesis, Chung Yuan Christian University, 2003.

[Chaturvedi, 02] Chaturvedi, D.K., Chauhan, R., Kalra, P.K.: Application of Generalized Neural Network for Aircraft Landing Control System, Soft Computing, vol. 6, pp. 441-118, 2002.

[Chiang, 07] Chiang, C.T., Chiang, T.S.: A Converged Recurrent Structure for CMAC_GBF and S_CMAC_GBF, Proc. IEEE International Symposium on Industrial Electronics, pp. 1876-1881, June 2007.

[Chiang, 05] Chiang, C.T., Chong, C.M.: Hardware Implementation of a Simple Structure of Addressing Technique for CMAC_GBF, Proceedings of the IEEE International Symposium on Industrial Electronics, pp.139-144, June 2005.

[Chuang, 07] Chuang, C.C., Hsu, C.C.: Integration of CMAC-GBF and Support Vector Regression Techniques, Proc. IEEE International fuzzy systems conference, 2007, pp.1-6, July 2007.

[Cohen, 95] Cohen, C.E. et al.: Automatic Landing of a 737 Using GNSS Integrity Beacons, Proc. ISPA, 1995.

[DDC-I, 95] DDC-I: Advanced Auto Landing System from Swiss Federal Aircraft Factory, Real-Time Journal, Sprint, 1995.

[FAA, 97] Federal Aviation Administration, "Automatic Landing Systems," AC20-57A (1997).

[FSF, 06] Flight Safety Foundation-Taiwan, http://www.flightsafety.org.tw/news1.php?Code=1 &main_id=3&pages=5, Flight International 10-16, Jan. 2006.

[Iiguni, 98] Iiguni, Y., Akiyoshi, H., Adachi, N.: An Intelligent Landing System Based on Human Skill Model, IEEE Transactions on Aerospace and Electronic Systems, vol. 34, no. 3, pp. 877-882, 1998.

[Izadi, 03] Izadi, H., Pakmehr, M., Sadati, N.: Optimal Neuro-Controller in Longitudinal Autolanding of a Commercial Jet Transport, Proc. IEEE International Conference on Control Applications, CD-000202, pp. 1-6, Istanbul, Turkey, 2003..

[Jorgensen, 97] Jorgensen, C.C., Schley, C.: A Neural Network Baseline Problem for Control of Aircraft Flare and Touchdown, Neural Networks for Control, pp. 403-425, 1991.

[Juang, 04] Juang, J.G., Chin, K.C.: Intelligent Landing Control Based on Neural-Fuzzy-GA Hybrid System, Proc. IEEE International Joint Conference on Neural Networks, vol. 3, pp. 1781-1786, 2004.

[Juang, 08] Juang, J.G., Lin, W.P.: Aircraft Landing Control Based on CMAC and GA Techniques, Proc. of IFAC WC 2008, P1730, Aug. 2008.

[Kaufmann, 95] Kaufmann, D.N., McNally, B.D.: Flight Test Evaluation of the Stanford University and United Airlines Differential GPS Category III Auotmatic Landing System, NASA Technical Memorandum 110355, June 1995.

[Liang, 00] Liang, Q., Mendel, J.: Interval Type-2 Fuzzy Logic Systems: Theory and Design, IEEE Transactions on Fuzzy Systems, vol. 8, no. 5, pp. 535–550, 2000.

[Liu, 07] Liu, Z., Zhang, Y., Wang, Y.: A Type-2 Fuzzy Switching Control System for Biped Robots, IEEE Transactions on Systems, Man, and Cybernetics—Part C, vol. 37, no. 6, pp. 1202-1213, 2007.

[NASDAC, 00] NASDAC Review of NTSB Weather-Related Accidents, http://www.nasdac.faa.gov/ aviation_studies/weather_study/studyindex.html, 2000.

[Wang, 04] Wang, C.H., Cheng, C.S., Lee, T.T.: Dynamical Optimal Training for Interval Type-2 Fuzzy Neural Network, IEEE Transactions on Systems, Man, and Cybernetics—Part C, vol. 34, no.3, pp. 1462-1477, 2004.