TECHNICAL NOTE

DESIGN AND SIMULATION OF AUTONOMOUS LANDING PHASE FOR AN UNMANNED AIR VEHICLE

K. Senthil Kumar* and J. Shanmugam**

Abstract

This paper deals with the design and simulation of autonomous landing phase for an unmanned Air Vehicle. Pitch angle and airspeed autopilots are developed, followed by glide slope and flare controllers to guide the aircraft to safe landing. During transition from glide slope to flare path, an UAV tend to be in the unstable region. A blending function has been formulated for use in UAV to overcome this unstability during transition. The flight path command simulation architecture is developed to provide the appropriate command signals for glide path, flare, blending and braking after touchdown. The flight path command signals with reference to the destination airport allows the simulation to perform at any designated airport supported by simulation graphics. Successful landings are demonstrated at a variety of simulated airports all over the world. The MATLAB / SIMULINK toolbox is used as a design tool.

Keywords*: Simulation, UAV, Autopilot, Landing Phase, Flight Control*

Introduction

The ultimate aim of both military and commercial aviation is all-weather operation. To achieve this goal, it should be possible to land the aircraft without visual reference to the runway. This can be accomplished by a landing system which will guide the aircraft down to a predetermined glide slope and then at a pre-selected altitude reduce the rate of descent and cause the aircraft to flare out and touch down with an acceptably low rate of descent [1]. The landing system will enable the UAV to land on a landing field. One of the very long term goals of the simulation project is to achieve fully autonomous flight and before this autonomous waypoint navigation simulations had been developed. The development of an autonomous landing simulation completes the requirements to begin fully autonomous flight simulation.

The Aerosim Flight Simulator employs a six degree-of-freedom (6 DoF) mathematical model with table lookup of aerodynamic coefficients for simulation of an UAV. It utilizes commercial gaming software X-Plane for graphics and includes an airport selector function that allows the simulation to operate at many airports supported by X-Plane. The simulator uses two coordinate systems, standard xyz and latitude-longitude-altitude (LLA). X-Plane requires LLA for graphical output.

The landing portion of autonomous flight is accomplished with reference to the destination airport. It also depends on the range from the aircraft to the runway. Because the airport is the main reference for flight path geometry and X-Plane uses LLA coordinates, these coordinates should be used to determine the range [2]. The requirements to successfully complete an autonomous landing are: define the glide path and flare path geometry, design autopilots for pitch, roll, and yaw, and design controllers for glide path, flare and directional course. In this study only longitudinal motion (motion in a vertical plane) is considered during the landing. Lateral motion is required primarily to point the aircraft down the runway and it is assumed that most of it is accomplished prior to the landing.

Sequence of Landing System

The profile of an automatic approach, flare and landing sequence is illustrated in Fig.1.When the aircraft has descended to 1500 feet radio altitude, the localizer and glide slope beams are captured .The localizer and glide slope

* Lecturer
Division of Avionics, Madras Institute of Technology, Anna University, Chromepet, Chennai-600 044, India
Email : ksk_mit@rediffmail.com
** Principal, Velammal Engineering College, Chennai-600 066, India
Manuscript received on 11 Jun 2007; Paper reviewed, revised and accepted on 25 Aug 2008

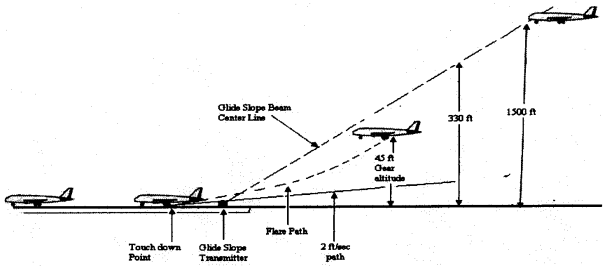

Fig.1 Automatic landing system

beam signals control the aircraft about the roll and pitch axes so that any deviations are automatically corrected to maintain alignment with the runway. At a radio altitude of 330 feet, the aircrafts horizontal stabilizer is automatically repositioned to begin trimming the aircraft to a nose-up attitude. The elevators are also deflected to counter the trim and to provide subsequent pitch control in the trimmed aircraft. When an altitude is reached at which the landing gear is 45 feet above the ground the flare mode is automatically engaged. The flare mode takes over pitch attitude control from glide slope and generates a pitch command to bring the aircraft onto a 2 feet/second descent path. At the same time, a throttle retard command signal is supplied to the auto throttle system to reduce engine thrust to the limits compatible with the flare path.

Glide Slope Control System

The first step in the design of autonomous landing is to define the glide path and flare path geometries [3]. The glide path is defined as a line from some starting point to the end of the runway. For this study, a glide path angle of -2.5° is used, so the starting point is defined by the LLA position of the end of the runway and the desired final approach distance, 23,000 ft in this case.

A real autonomous landing controller would rely on signals emitted from stations at the end of the runway to maintain the appropriate glide path. As a result, the controller becomes more sensitive as the aircraft gets closer to the runway threshold. To simulate this relation of the range from the aircraft to the runway, the glide path command signal is defined to include the range. Fig.2, shows the glide path geometry where the commanded height above ground is a function of the range [4].

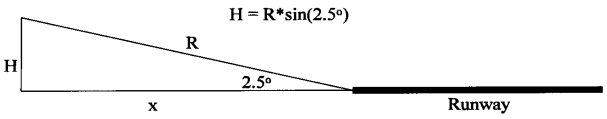

Fig.2 Glide path geometry

Design Objectives : Landing the aircraft requires a complex combination of δ_e and δt to coordinate the speed/pitch/height control. The UAV is to be controlled to a height of, approximately 1200 feet and a range of 23,000 feet from the touch down point. The UAV is to be guided through the specified glide slope of 2.5° under no wind as well as specified wind conditions [5].

Certain limits are imposed on the dynamic variables, which are,

$$-7^{o} \le \alpha \le +7^{o};\ -16^{o} \le \theta \le 16^{o};$$
$$-20^{o} \le (\delta_e, \delta_a, \delta_r) \le +20^{o} \quad (1)$$

Servo rate ≤ 100° / sec. The Elevator deflection (δ_e) and throttle are the plant input and Pitch attitude (θ), deviations from the glide path (d) are the plant output.

To simulate, the increasing sensitivity to range, the error signal should be defined as the error in glide slope angle γ. As shown in Fig.3, the airborne glide path receiver

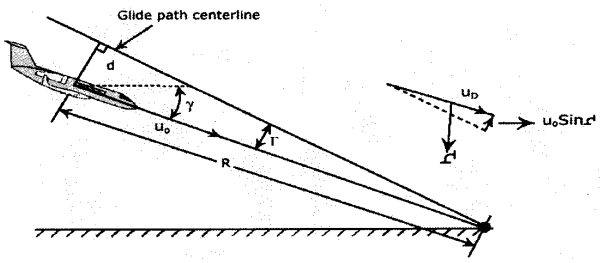

Fig.3 Geometry of longitudinal control

will measure the angle of error between the glide slope and the UAV [1], [3]. This angular error will then be converted to longitudinal deviation (d) as

$$d = R \sin(\delta) \quad (2)$$

The aim is to control d; once the condition of d = 0 achieved, the UAV follows the glide slope line. Now,

$$\dot{d} = U \times \sin(\gamma - \gamma_{ref}) \quad (3)$$

and for small angles,

$$\dot{d} = U \times (\gamma - \gamma_{ref}) \quad (4)$$

Integration of the above equations gives the information about the longitudinal deviations, which is fed to the

controller, and then the appropriate control input is computed. If d is not directly measurable, this can be written as $\sin\Gamma = d/R$ and can be feedback on Γ, which is measurable from ILS shown in Fig.4.

Using this relationship, the angular error can be found in terms of the range where for the same linear error, the angular error becomes increasingly larger as the range gets smaller. At some point, the sensitivity of the glide slope controller exceeds its ability to keep the airplane on the glide path. At this point, the flare controller, which is not dependent on the range, must take over control of the airplane.

The block diagram representation of the glide slope control system is shown in Fig.5. The pitch hold autopilot is the basic autopilot mode for this system to descent from a particular height. The glide slope controller has a coupler, which contains a PID controller and a lead compensator.

Flare Path Control System

In flare path region, the aircraft follows an exponential path. During the flare maneuver, pilots transition from flying a straight line to an exponential path to slow the descent rate of the airplane. This can be simulated by defining an exponentially decaying flight path and using altitude above ground to generate the error signal to the controller. Fig. 6. shows the flare path geometry with the intended touchdown zone approximately 500 ft. from the runway threshold.

The equation, which governs the idealized, exponential flare trajectory, is,

$$h = h_0 \, e^{-t/\tau}$$

The exponential decay constant τ is a function of the distance x and the distance to the touchdown zone from the threshold as exponential functions decay in approximately 5 decay constants. τ must be selected such that the

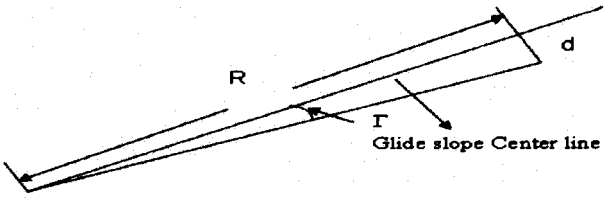

Fig.4 Range vs angular error

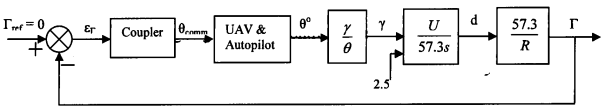

Fig.5 Glide slope control system

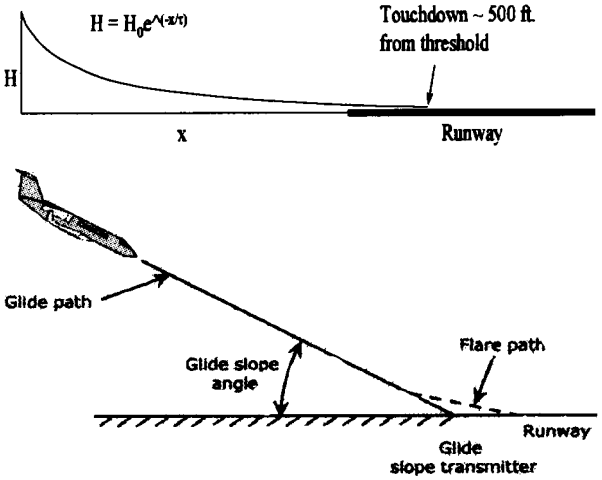

Fig.6 Flare path geometry

airplane touches down where desired and at an acceptable vertical speed.

The block diagram representation of the flare control system is shown in the Fig.7. The outer loop simply supplies the rate-of-descent command $\dot{h}_r$ .The pitch attitude hold autopilot is the basic autopilot mode for this system to descent from flare entry height h_o.

Upon main gear touchdown, the elevator must allow the nose of the airplane to rotate downward to contact the nose gear with the ground and the brakes must be applied [6]. Because exponentially decaying functions never actually reach zero, the elevator control command must be switched from the flare path to neutral upon main gear touchdown. The brakes must be applied smoothly after touchdown or the gear will fail. A rate limiter after a switch can be used to accomplish this.

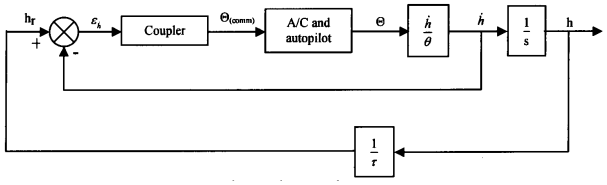

Fig.7 Flare path control system

With the flight path geometry defined, autopilots for pitch, yaw, and roll must be designed to fly the airplane autonomously. Then controllers for glide slope, flare, and directional course must be designed to keep the airplane on the desired flight path. Because of the complexity of the problem, it was determined that only pitch control should be used without wind disturbances to accomplish the landing. Upon successful landing in ideal conditions, lateral controllers can be designed for conditions with wind disturbances.

Range and Height Calculation

The next step in this design is to calculate the range to the runway threshold because the glide slope geometry and control signal error depend on the range to the runway threshold. The Airport Selector S-function is evaluated to determine if the outputs from it were suitable for determining the range. The initial LLA and initial distance from the runway are output from the Airport Selector function and are suitable for finding the initial range and runway threshold LLA. The Fig.8. illustrates the geometry to find out the instantaneous range of UAV which mainly depends upon the Latitude, Longitude and Altitude (LLA) of the UAV and the destination Runway.

The latitude and longitude are represented in degrees, where there are 60 minutes per degree and 60 seconds per minute and must be converted to feet so that they have the same units as altitude. The latitude conversion to feet is relatively constant from the equator to the poles and is approximated at all points as 6076 feet per minute. The longitude conversion to feet varies from the equator to the poles. This is because the lines of longitude become closer towards the poles. The circle created by intersecting a plane with the earth at some line of latitude will have a radius equal to the radius of the earth times the cosine of the latitude angle. The radius of this circle is used to calculate the circumference of the earth at that particular latitude. Regardless of the circumference of the circle, it still contains 360 degrees, thus a conversion factor can be found. The average radius of the earth is 36,522 feet [7]. This then yields the conversion factor for longitude as 36,522*cos(latitude^{o}) feet per minute.

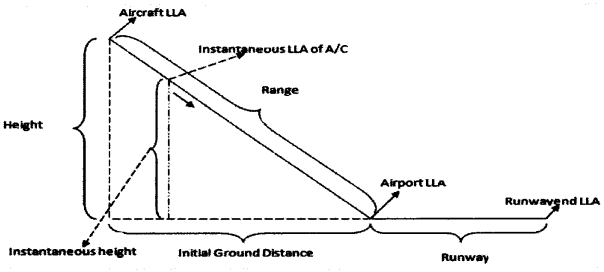

Fig.8 LLA calculation

Steps (i-v) are used to calculate the LLA [8].

i) Known parameteres are airport latitude, airport longitude, Base elevation of the Runway and Initial ground distance from which glide path starts.

ii) Based on Range, Height of aircraft above base elevation are calculated.

The symbol and description of the various parameters used in the equations are given in below :

Symbol	Description
Gd_{ini}	Initial ground distance
$lat_{Initial}$	Initial latitude
$alt_{Initial}$	Initial altitude
$lon_{Threshold}$	Threshold longitude
$lat_{Runwayend}$	Runwayend latitude
$alt_{Runwayend}$	Runwayend altitude
$range_{Delta}$	Change in range
$h_{Initial}$	Initial height
T_{head}	True heading
$lon_{Initial}$	Initial longitude
$lat_{Threshold}$	Threshold latitude
$alt_{Threshold}$	Threshold altitude
$lon_{Runwayend}$	Runwayend longitude
$range_{Initial}$	Initial Range
$range_{ins}$	Instanteneous height
h_{ins}	Instantaneous height

iii) The aircraft LLA is calculated as

$$Gd_{ini} \times \cos\left(\frac{\pi}{180} \times T_{head}\right) \times \frac{1}{6076 \times 60} + lat_{initial} = lat_{Threshold} \tag{5}$$

$$\frac{Gd_{ini} \times \sin\left(\frac{\pi}{180} \times T_{head}\right) \times \frac{1}{6076 \times 60}}{\cos\left(\frac{\pi}{180} \times lat_{Initial}\right)} + lon_{initial} - lon_{Threshold} \tag{6}$$

$$\text{Base elevation} = \text{alt}_{\text{Threshold}} \tag{7}$$

$$\text{alt}_{\text{Initial}} = \text{h}_{\text{Initial}} + \text{Base elevation} \tag{8}$$

iv) The Runway end LLA is claculated as

$$\left[\left[\left\{\text{Gd}_{\text{ini}} + \text{Runway length}\right\} \times \cos\left(\frac{\pi}{180} \times \text{T}_{\text{head}}\right) \times \frac{1}{6076 \times 60}\right] + \text{lat}_{\text{Initial}}\right] = \text{lat}_{\text{Runwayend}} \tag{9}$$

$$\left[\frac{\left[\left\{Gd_{ini} + \text{Runwaylength}\right\} \times \sin\left(\frac{\pi}{180} \times \text{T}_{\text{head}}\right) \times \frac{1}{6076 \times 60}\right]}{\cos\left(\frac{\pi}{180} \times \text{lat}_{\text{Initial}}\right)} + \text{lon}_{\text{initial}}\right] = \text{lon}_{\text{Runwayend}} \tag{10}$$

$$\text{Base elevation} = \text{alt}_{\text{Runwayend}} \tag{11}$$

v) Instantaneous Range and height is calculated as

$$\sqrt{\begin{array}{l}\left[\left[\text{lat}_{\text{Threshold}} - \text{lat}_{\text{Initial}}\right] \times 6076 \times 60\right]^2 + \\ \left[\left[\text{lon}_{\text{Threshold}} - \text{lon}_{\text{Initial}}\right] \times 6076 \times 60 \times \cos\left(\frac{\pi}{180} \times \text{lat}_{\text{Initial}}\right)\right]^2 + = \text{range}_{\text{initial}} \\ \left[\text{alt}_{\text{Threshold}} - \text{alt}_{\text{Initial}}\right]^2\end{array}} \tag{12}$$

$$\sqrt{\begin{array}{l}\left[\left[\text{lat}_{\text{Ins}} - \text{lat}_{\text{Initial}}\right] \times 6076 \times 60\right]^2 + \\ \left[\left[\text{lon}_{\text{Ins}} - \text{lon}_{\text{Initial}}\right] \times 6076 \times 60 \times \cos\left(\frac{\pi}{180} \times \text{lat}_{\text{Initial}}\right)\right]^2 + = \text{range}_{\text{Delta}} \\ \left[\text{alt}_{\text{Ins}} - \text{alt}_{\text{Initial}}\right]^2\end{array}} \tag{13}$$

$$\text{range}_{\text{Initial}} - \text{range}_{\text{Delta}} = \text{range}_{\text{Ins}} \tag{14}$$

$$\text{alt}_{\text{Ins}} - \text{Base elevation} = \text{h}_{\text{Ins}} \tag{15}$$

Sample Calculation

In this paper, the Dallas Fort Worth International airport has been considered for the landing phase of UAV with the following known parameters: True heading of 180.3°, Base elevation of 607 feet, Latitude of 2.93483568652165°, Longitude of -97.0268825156042° and Initial ground distance of 24300 feet. These values are used to calculate initial range, instantaneous range, initial height and instantaneous height. Using steps (iii) to (v) the calculated values are found to be:

Initial height = 1060.960 ft
Initial altitude = 1667.96 ft
Initial latitude = 33.00149046735776 deg
Initial longitude = - 97.0272315271941 deg
Runwayend altitude = base elevation = 607 ft
Runway length = 6076 ft
Runwayend laititude = 32.91816924831753 deg
Runwayend longitude = - 97.0267952483441 deg
Initial range = 24323.1502096 ft
Delta range = 6361 ft
Instantaneous range = 24323.1502096 -6361 = 17962 ft

With simulation run in Matlab /Simulink environment for about 20 sec, it is observed that the simulated instantaneous range is equal to the calculated instantaneous range. Hence, it has been proved that the algorithm is more efficient and this has been verified with many more airports around the world.

Autopilot Design Procedure

To begin with the design of the pitch angle autopilot, a transfer function representative of the UAV in landing conditions is required. The table used [9],[10],[11] by the 6DoF S-function lists the weight and lift coefficient. Using this value, and the desired speed, the angle of attack and Mach number could be found. Knowing the angle of attack and Mach number, the drag coefficient and pitching moment coefficient are determined. With these values and the other stability and control derivatives, a transfer function for pitch angle to elevator deflection could be determined from the short period mode literal factors approximation. A script is written in MatLAB to find transfer functions for short period, dutch roll, and spiral modes by literal factors and generated the transfer function (17) is used for pitch angle autopilot design.

$$\frac{\theta}{\delta_e} = \frac{-5.329 \times 10^{-15} s^4 + 35 s^3 + 223.6 s^2 + 17.45 s}{s^5 + 8.804 s^4 + 56.4 s^3 + 4.327 s^2 + 0.927 s} \tag{17}$$

Using MatLAB's SISO tool, the compensator is designed [12] for pitch angle to yield a damping ratio of 0.709, a crossover frequency of 4.68 rad/s, and a phase margin of 67.6 degrees. Fig.9 shows the SISO tool display for the pitch angle autopilot. The root locus shows a stable system with damping of 0.709, so approximately three

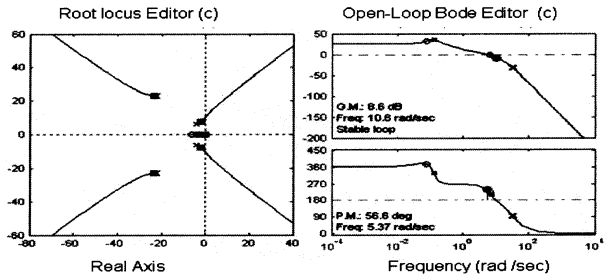

Fig.9 Root locus and bode plot for pitch angle autopilot

overshoots are expected in the response. The Bode plot shows a crossover frequency of 4.68 rad/s which indicates the system should respond well for the high-bandwidth task of landing, but will not respond to high-frequency disturbances and noise.

Figure 10 shows the predicted and actual responses of the pitch angle autopilot to a step command. The predicted response shows three overshoots as expected and a suitable settling time of four seconds. The actual response, however, is radically different. The initial response is opposite to the command, and then with some oscillation and one overshoot, the aircraft reaches the desired pitch angle with a settling time of approximately 12 seconds.

The actual settling time of 12 seconds is much slow for an autopilot to control an aircraft on landing. It is observed that the plant from which the autopilot was designed is obviously much different than the actual simulation. Two possibilities for explanation exist. Either the mathematical calculations within the 6DoF S-function are not correct or the literal factors approximation is not valid for this case. Regardless, another method is required to design an appropriate autopilot for pitch angle.

Two methods were used during the initial design stages:

- Design of a lead-lag compensator to achieve the desired response
- Tuning a PID controller

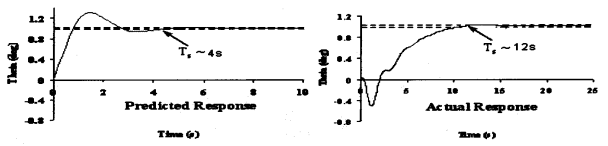

Fig.10 Predicted and actual responses of the pitch angle autopilot to a step command

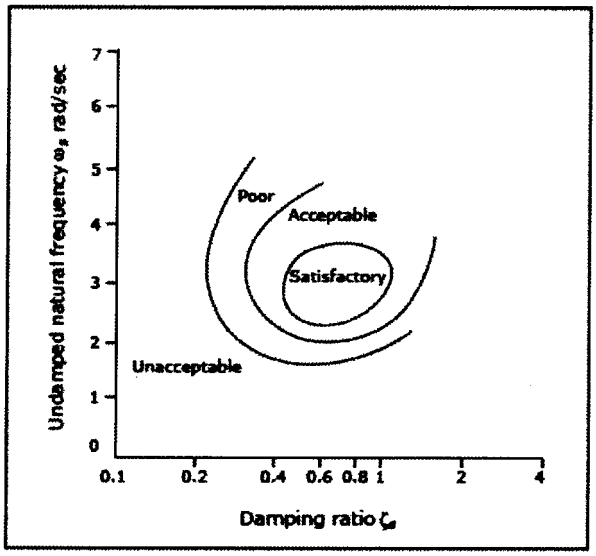

Fig.11 Natural frequency - damping criteria

Pole placement for lead-lag was attempted, but abandoned in favor of using a PID controller as there was no progress. An iterative technique was used to obtain better performance of the pitch controller. Conventional root locus based design methodology is followed. Fig.11 gives a plot for the expected dynamic behavior of an aircraft. These results are based on a survey which determined the acceptable natural frequencies and damping ratios for the aircraft such that pilots felt comfortable handling of the aircraft.

From theoretical analysis, it is observed that the system poles lie as shown in Fig.12 and that the desired locations of these poles lie in the shaded region. By using the root

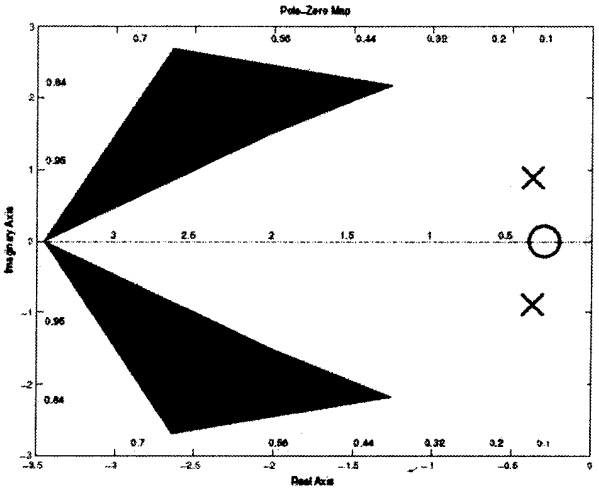

Fig.12 Actual and desired location of poles

locus, the inner Pitch rate loop gain was determined by selecting the maximum damping available. The outer Pitch angle control loop PID gains were selected by iteration. The gain was first adjusted such that the closed loop poles are in the desired area of the s-plane. Using linear simulation, the result was observed. The control effort required and the steady state behavior of the response was noted. As the actuator deflections were beyond saturation, the Proportional gain was reduced. Integral gain was added based on the steady state error values. If the overshoot produced was greater than 20%, the integral gain was reduced. Finally, the derivative gain was adjusted to obtain the shortest possible settling time. The resultant controller was then simulated using a nonlinear model. Fig.13 shows the actual time response of this system to a step command. The settling time is just less than six seconds with one overshoot and some minor oscillation.

In order to design the pitch angle autopilot adequately for glide slope control, airspeed control is required. The engine model currently running in the simulation has very little time lag. In fact, it is very small so it can be negligible. So, it is assumed that a lag compensator would be required to maintain airspeed. The desired airspeed is 330 ft/s and the controller uses forward airspeed feedback. However, the throttle command signal must be a small number between zero and one where zero is reverse thrust, 0.1 is neutral, and one is maximum forward thrust. So, to normalize the error signal to a maximum value of one, a gain of 1/330 is imposed. The initial assumption of lag compensation proved correct and with little effort, the compensator (18) is found to provide exceptional performance.

$$\frac{3(s+1)}{(s+.5)} \qquad (18)$$

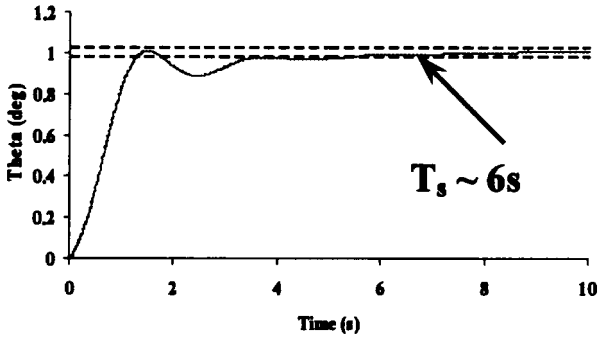

Fig.13 Actual response of pitch angle autopilor with PID compensation

The design of glide slope controller is not so complicated as compared to the pitch autopilot. The two methods that are considered for designing an adequate controller are attempt to place lead-lag poles and zeros and search for PID gains. On the assumption that lead-lag should work for this design, the first method is employed. From the Fig.14 the Root Locus and Bode plots of the uncompensated open-loop transfer function, it was evident that the system was only stable at low gains, which made it extremely difficult to get a descent response time from the closed-loop transfer function given no compensation. The Magnified View of the Root Locus Plot of the System without a Compensator is shown in Fig.15.

As a result, a compensator was designed using the Root Locus and Bode plots that attempted to make the system stable with good phase margin, had high gain at low frequency and low gain at high frequency and had reasonable bandwidth and crossover frequency.

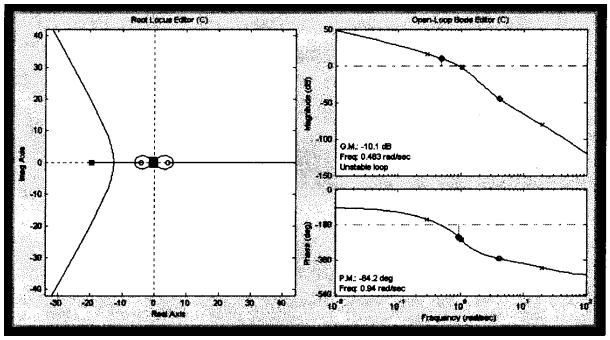

Fig.14 Root locus and bode plot of the system w/o a compensator

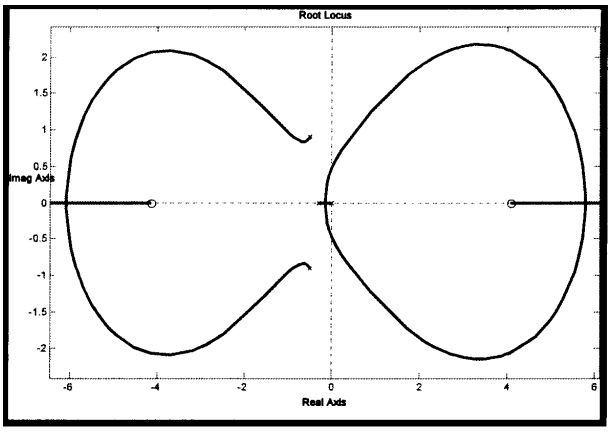

Fig.15 Magnified view of the root locus plot of the system w/o a compensator

Then, experimentally, the compensator (19) is determined to provide good performance.

$$\frac{50\,(s+.5)\,(s+.3)}{(s+1)\,(s+1)} \qquad (19)$$

Magnified View of the Root Locus Plot of Compensator is shown in Fig.16. As gain increases, the critical pole(s) start near the origin and move further to the left until they intercept the ellipse at around the (-3) value on the real axis and then split and trace the ellipse back until it intercepts the imaginary axis and goes unstable. The advantage of this compensator is that it allows more range of gain to make the system unstable. As a result, higher gain at low frequency is more attainable with this compensator.

This is shown in Fig.17. Where the airplane falls below glide path initially, but comes back with no overshoot while still more than 10,000 ft. from the runway.

After experiencing the difficulty of trial and error design for the glide slope controller, the flare controller design would be similar. Using MatLABs SISO tool, root locus and bode plot of flare path control system is shown in Fig.18.

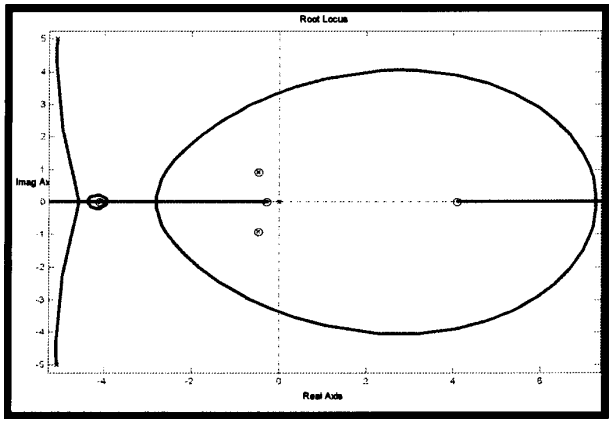

Fig.16 Magnified view of the root locus plot of compensator

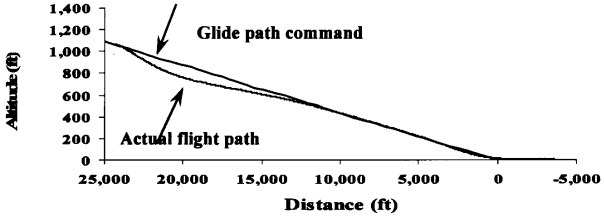

Fig.17 Actual glide slope response

Results and Discussion

The glide slope controller performs well in updrafts and downdrafts as shown in Fig.19. In the updraft case, the aircraft virtually does not leave the desired path until the flare command and in the downdraft case, the aircraft is still established on the glide path at a range of around 10,000 ft.

However, there is no way to start the simulation at the flare engagement point, so the method of switching control from the glide slope to the flare controller had a profound effect on the ability to design the flare controller. Blakelock [1] suggests that the flare controller should be the same as the glide slope controller with an additional lead network. After completing the glide controller an attempt for the flare controller was made.

Blakelock[1] also suggests that the sink rate should be controlled. This is because the flare path command is an exponential function for which the derivative is a constant multiplied by the instantaneous altitude. The constant is the inverse of the exponential decay time constant. Then, assuming that the aircraft will touch down in four or five

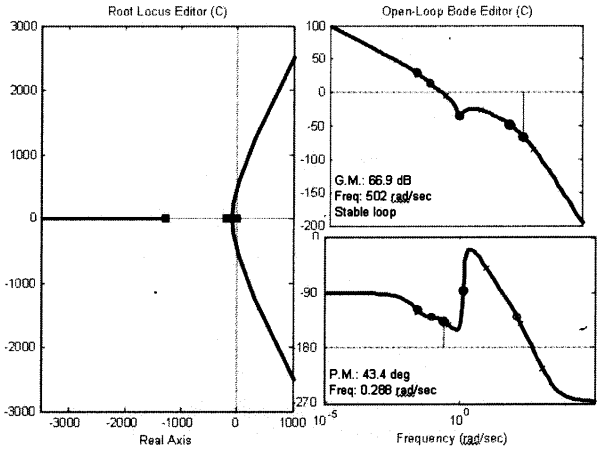

Fig.18 Root locus and bode plot of flare path control system

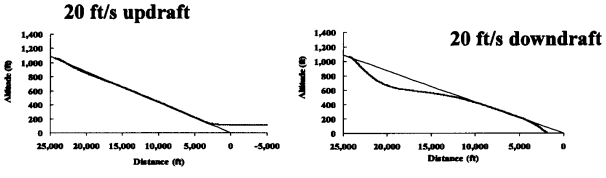

Fig.19 Glide slope response with updraft and downdraft of 20 ft/s

time constants, the appropriate time constant for the function can be determined. The following equations show this relationship for the flare path command.

$$H = H_0 \, e^{-t/\tau} \tag{20}$$

$$\dot{H} = -\frac{1}{\tau} H_0 \, e^{-t/\tau} \tag{21}$$

$$\dot{H} = -\frac{1}{\tau} H \tag{22}$$

Considering Blakelocks [1] suggestions, the aircraft sink rate is used for the flare command and along with the glide slope controller in the initial attempt. The range at glide slope/flare switch is selected as 2,000 ft. This is because the glide slope controller had proven to perform well to a range somewhat less than 2,000 feet and the aircraft is just under 100 feet altitude above ground level at a range of 2,000 ft. The initial attempt is to switch the signals at the instant the range became equal to or less than 2,000 ft. A problem is immediately evident, however as the aircraft became amazingly unstable at the moment of command signal switch.

Upon further evaluation of Blakelocks methods, it is determined that a time dependent function is not appropriate as it required that the aircraft be at the proper position for flare at a certain time step in the simulation. There is no way to guarantee this requirement, so the time dependent exponential function is abandoned. The geometry of the glide path and flare are both dependent on the position of the runway threshold. Defining the flare path command in terms of the distance of the runway seemed much more appropriate because the distance of the runway is always known, regardless of the simulation time step. The path is then defined as:

$$H = H_0 \, e^{-x/\tau} \tag{23}$$

where x is the distance from the glide slope engagement point determined by subtracting the instantaneous LLA from the initial LLA, converting the latitude and longitude differences to feet and solving for x by the Pythagorean Theorem. Because the starting point of the flare command signal coincides with the glide path command signal, the value of the decay constant is found to be 5,500. This corresponds to touchdown in approximately five decay constants.

Having the flare path defined in terms of the distance from the runway, the altitude above ground level is used as feedback to generate the error signal. Again the glide slope controller is evaluated for use in the flare, but not suitable as the command signal is no longer dependent on the sink rate. By experiment, it is observed that with a very small gain, the aircraft could be made to flare very well and touch down at any point desired. With a flare command gain of 0.0004995, the aircraft touches down approximately 500 ft. after the runway threshold. The only problem with this is that it requires ideal conditions to work as there is no integrator in the compensator.

An additional problem in the flare controller design is the method of switching from glide slope to flare command signals. With the switch at a range of 2,000 ft., the aircraft could not be made stable as having a switch in the simulation caused adverse affects on the very input signals to the switch. It was determined by experiment that constants in Simulink create problems in solving the algebraic loop and cause the simulation to produce erroneous results. To compensate, step blocks were used with the step value equal to that of the desired constant and the step time equal to the first time step of simulation (0.01 s). Even with this correction, stability problems are still evident at the switch of control commands.

To compensate for the sudden switch in commands, a blending function is developed to soften the effect of the switch. This function blends the signals over range values of 5,000 ft. to 3,000 ft. These values are selected to ensure that the aircraft would be established under the flare controller before reaching the desired switch range of 2,000 ft.

Design of Blending Function

The blending function [8] is mixing of signals during the transition from glide path to flare path. This function is conceived in order to solve the problem of extreme oscillations and instability during the transition period. The proposed geometry of blending function is shown in Fig.20. In the blending function the gain of glide and flare path is varied according to variation in range. It is found that the glide path gain is decreasing and the flare path gain [8] is increasing. By using a limiter, the upper limit of the gain is set to 1 and the lower limit is set to 0. At any point sum of the glide and flare path gains is 1. From range R_1 to R_3 the glide path alone will be present, the gain of the glide path is 1 and the flare path gain is 0. From R_2 to 0 only the flare path will be present, the glide path gain is 0

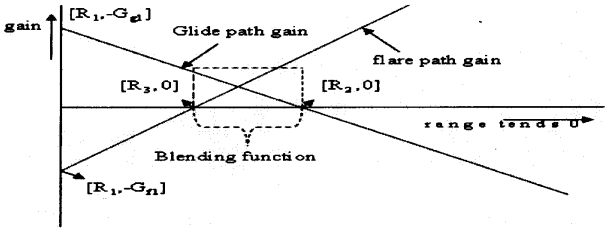

Fig.20 Geometry of blending function

and the flare path gain is 1. In between R_3 and R_2 the blending function will occur and the gain will vary with glide path gain decreasing from 1 to 0 and flare path gain increasing from 0 to 1. The condition of the range is ($R_1 > R_3 > R_2 > 0$).

R_1 - The point from which glide path starts
Gg_1 - Gain at which glide path starts
R_2 - The range at which glide path gain becomes zero
R_3 - The range at which flare path gain becomes zero and
G_{f1} - Flare path gain at R_1

The equation of straight line with coordinates (x_1, y_1) and x_2, y_2) is given by

$$(y - y_1) = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) \tag{24}$$

such that the glide path gain equation with the coordinates (R_1, G_{g1}) and (R_2, 0) as

$$G_g = \frac{G_{g1}}{R_2 - R_1} (R_1 - R) + G_{g1} \tag{25}$$

Where G_g - Glide path gain at every instant of Range R.

$$G_f = \frac{G_{f1}}{R_3 - R_1} (R - R_3) + G_{f1} \tag{26}$$

Where G_f - flare path gain at every instant of Range R.

In the region between range R_3 and R_2 the blending phenomenon will occur.

Figure 21 shows the Simulink architecture [13] of the function. The saturation blocks normalize the signal multipliers to a value between zero and one. 3,000 is subtracted from the range and multiplied by a gain to generate a multiplier for the glide path command. The gain is selected such that at 5,000 ft. range, the glide multiplier is

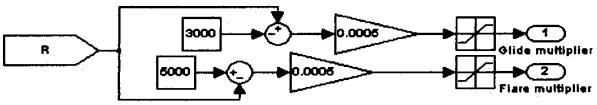

Fig.21 Glide slope/flare blending function

one and at 3,000 ft. range, the glide multiplier is zero. The flare multiplier is much the same, but in the reverse direction. These multipliers directly multiply the glide and flare signals such that when the range is between 5,000 ft. and 3,000 ft., both signals are active. Above 5,000 ft. only the glide path signal is active and below 3,000 ft. only the flare signal is active.

Response of the Parameters Variation with Blending Function

The responses of the various parameters with the blending function are shown in Figs. 22-27. The blending phenomenon occurs in the range 5000ft to 3000ft, which means that between 5000ft and 3000ft the UAV will be in both glide path and flare path. Above the altitude of 5000ft (i.e. from 23000ft to 5000ft) UAV will be in the glide path and below the altitude of 3000ft (i.e. from 3000ft to touchdown) it will be in the flare path only. In terms of time, the blending phenomenon will occur between 59 and 65 seconds. During 0-59 seconds only the glide path will be present and after 65 seconds only the flare path will be present. From the Fig.27 it is inferred that the airspeed autopilot is maintained during landing. At the end of the landing phase the Engine rpm, throttle position and thrust are getting reduced.

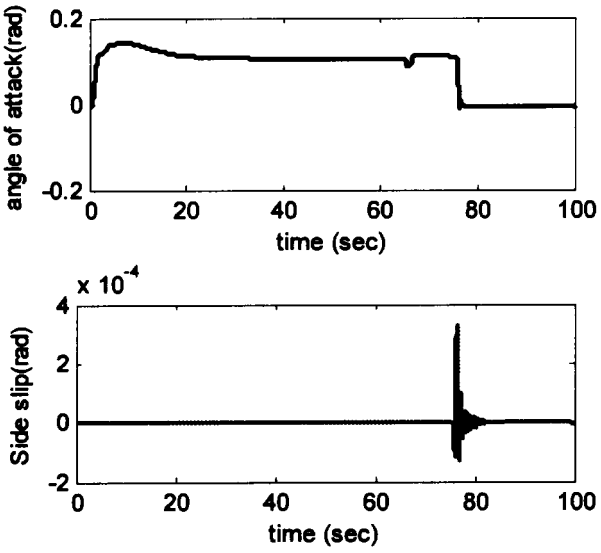

Fig.22 Response of angle-of attack and sideslip

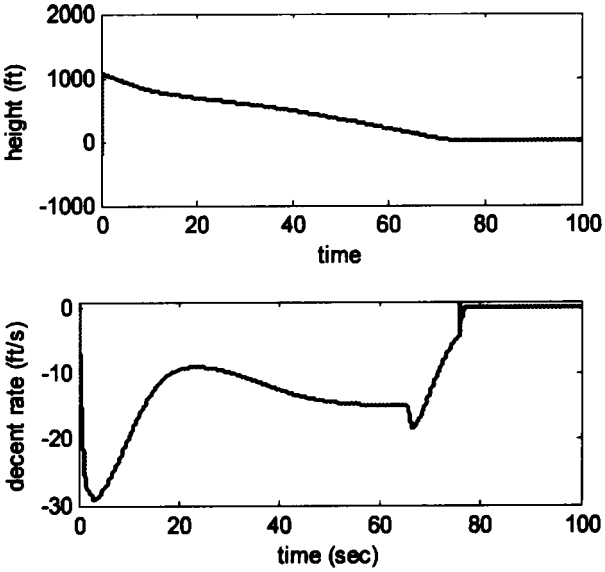

Fig.23 Response of height and decent rate

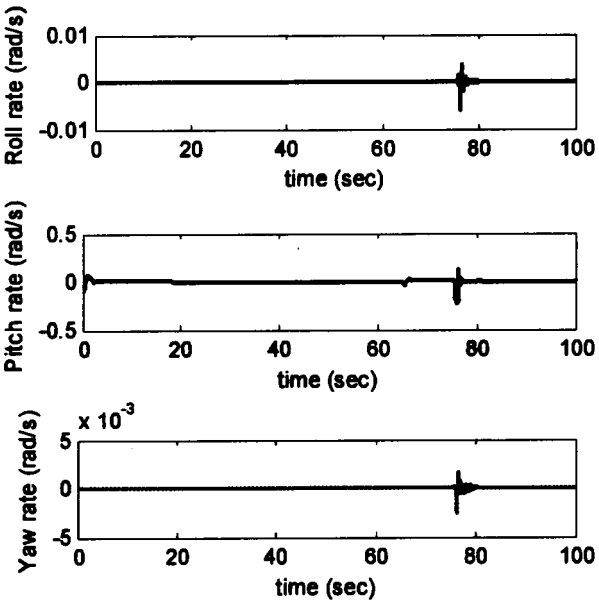

Fig.25 Response of pitch rate, yaw rate and roll rate

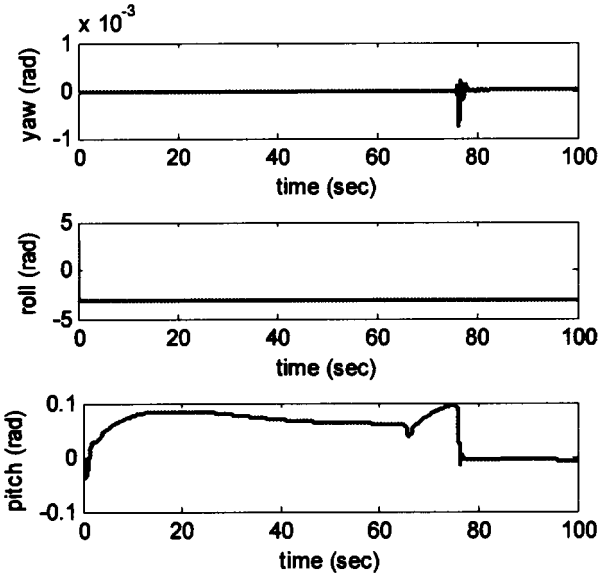

Fig.24 Response of pitch, roll and yaw angles

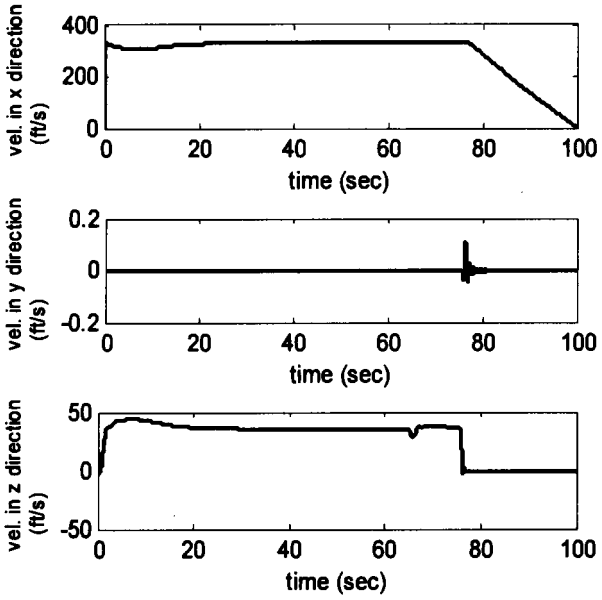

Fig.26 Response of Aircraft velocities in x, y and z directions

Comparison of Performance with and without Blending Function

The Figs.28-35 show the comparison of angle of attack, sideslip, height, pitch rate, roll rate and yaw rate with and without blending function. The dotted lines indicate the parameter variation with blending function and thick line indicates the parameter variation without blending function. The parameters are compared with range variation.

In the range from 5000 ft to 3000 ft with time variation from 59 to 65 seconds, the angle of attack variation with and without blending function is as shown in Fig.28. On comparison, it is found that when the blending function is not included, the variation is large i.e. around 15% but

when the blending function is included the variation is reduced to about 3%.

The decent rate shown in Fig.29 is smoother with the blending function than when compared to without the blending function. The smooth variation of decent rate means the oscillations get reduced and the steepness will also get reduced.

The comparison of height is shown in Fig.30 proves that the steepness of the aircraft is reduced. The exponential decay is found to be good while including blending function. If the steepness increases i.e. without blending function the force on the landing gear will also be increased. This may cause the landing gear failure and wear of tires which makes the landing of the aircraft difficult.

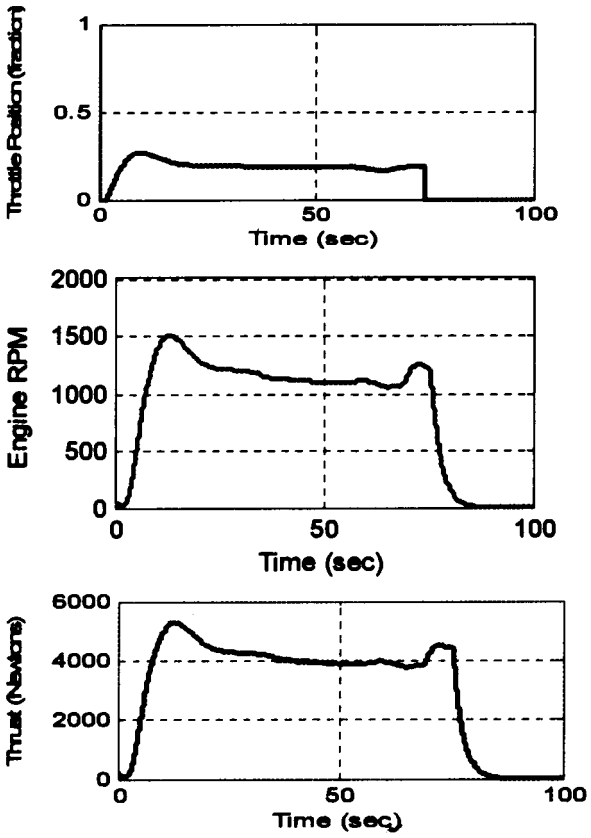

Fig.27 Response of engine rpm, throttle position and thrust

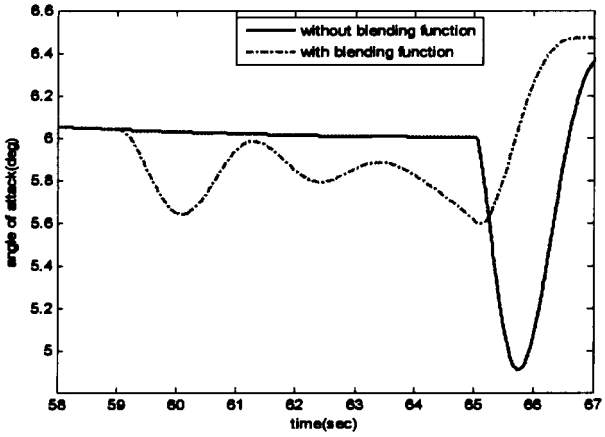

Fig.28 Response of angle of attack

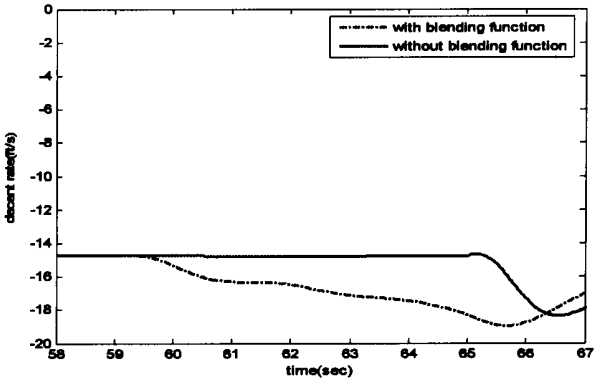

Fig.29 Response of decent rate

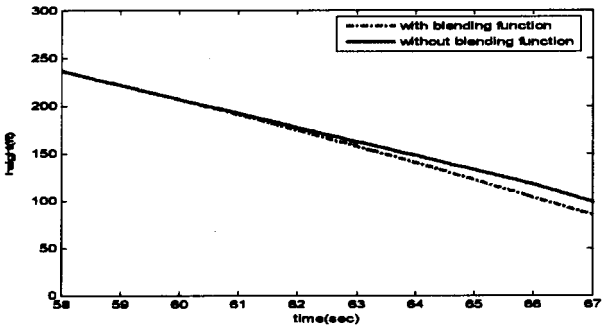

Fig.30 Response of height

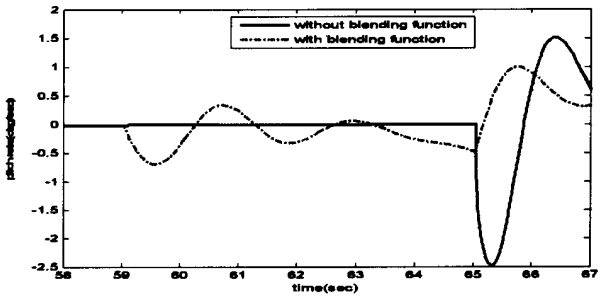

Fig.31 Response of pitch rate

On comparing the pitch rate as shown in Fig.31, it is found that the variation and the oscillations are considerably when the blending function is not included. The variations are reduced when the blending function is included.

The comparison of pitch is shown in Fig.32. It is found that the instant pitch is available by summing the pitch at the previous instant and the pitch rate at that instant. Hence, when the pitch rate varies the pitch will also vary automatically.

The variation of velocities with respect to x and z directions are shown in Fig.33 and Fig.34 respectively with and without the blending function. The blending function ensures reduced velocity in Z-direction which will cause smooth contact of the wheels with surface on touch down.

The elevator command which is given to the elevator is shown in Fig.35, with and without blending function.

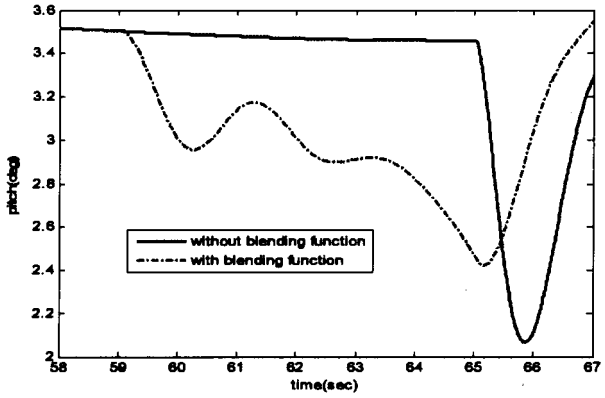

Fig.32 Response of pitch

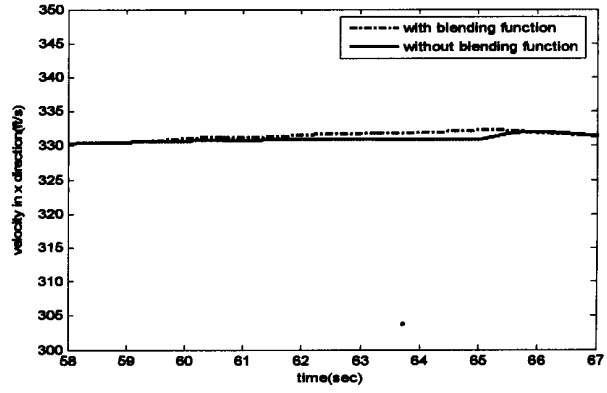

Fig.33 Response of velocity in x-direction

The elevator deflection seems to be more when the blending function is not used, requiring more control power which might cause damage to control surface resulting in fatal accidents during landing. By using the blending function, the control power required to move the elevator is substantially reduced.

Thus by using blending function the oscillations are reduced and the control power is also considerably reduced as evident from the graph and ensures safe landing.

Comparison of the Performance Measures

The typical values of the parameters during the various landing phases are given in Table-1. Measures of performance are required to specify the desired landing conditions of aircraft. Basically, they require that the aircraft must land within the desired envelope of dispersions. The Table-2 summarizes the blending performance measures.

The blending of signals at the switch of glide slope and flare control signals solved the problem of extreme oscillation and instability during the switch. In fact, graphically, the switch is almost unnoticeable. Fig.36 shows the

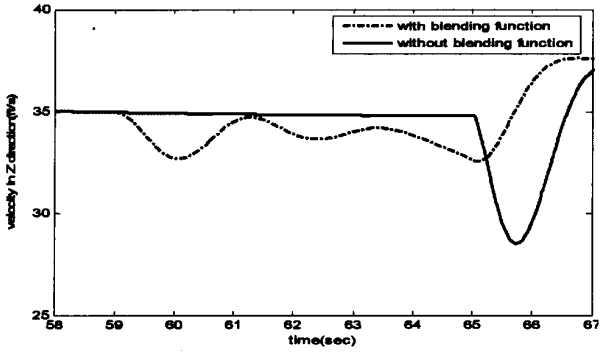

Fig.34 Response of velocity in z-direction

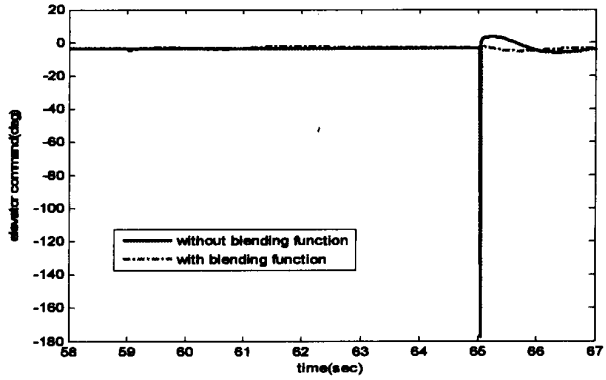

Fig.35 Response of elevator comand

Table-1 : Typical values of the parameters during landing				
Parameters	*At the Glide Slope begin (0 sec)*	*At the start of blending function (59 sec)*	*At the end of blending function (65 sec)*	*At touchdown point (86 sec)*
Angle of attack in degrees	2	6 deg	5.6 deg	-0.3 deg
Sink rate ft/s	20.8	14.71 ft/s	-18.6 ft/s	-0.001 ft/s
Altitude in ft	1074	221 ft	123 ft	7.5 ft
Pitch rate deg/s	3.7	-0.14 deg/s	-.5 deg/s	0 deg/s
Pitch angle in deg	0.5	3.5 deg	2.5 deg	-0.3 deg/s
Incremental Forward velocity (U) in ft/s	325	330 ft/s	332 ft/s	184 ft/s
Incremental vertical velocity (w) in ft/s	12	35 ft/s	32.6 ft/s	-1 ft/s
Throttle	0.045	-0.06	0.2466	0.1
Elevator command in deg	0.18	0.2696	-0.06 deg	0 deg
Range in ft	24330	5000	3000	0

Table-2 : Performance measures		
Parameter	*Without blending function*	*With blending function*
Angle of Attack	Variation is High	Variation is Low
Decent rate	Sudden Change	Smooth Change
Height	Steepness is High	Steepness is Low
Pitch Rate	Variation is High	Variation is Low
Pitch	Variation is High	Variation is Low
Forward Velcocity in X direction	Variation is not Smooth	Variation is Smooth
Forward Velocity in Z Direction	Variation is High	Variation is Low
Elevator Comand	High Deflection	Small Deflection

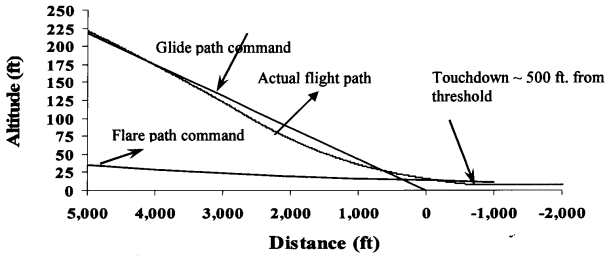

Fig.36 Aircraft response during glide slope/flare switch and touchdown

response of the aircraft during the signal blend. Oscillation is not visible on this graph, but is very slightly visible in the graphics produced by X-Plane [14]. The figure also shows the response of the aircraft during the flare. It flies an exponential path to touchdown approximately 500 ft. from the runway threshold. It also touches down at a vertical velocity of approximately four feet per second, which would be considered a good landing by any pilot.

Because the flare path command is exponential, the aircraft tends to bounce if the elevator remains under control of the flare controller after main gear touchdown. To ensure the aircraft remains on the ground, the elevator

is neutralized with a relay at an altitude above ground level equal to the height of the main gear (9 ft). At the same altitude, the throttle command is also neutralized and the brake command is changed from one to zero (zero to full braking) with a rate limiter to limit the brake application time to two seconds. The rate limiter on braking prevents gear failure due to over-braking upon touchdown.

Upon completion of a successful landing in ideal conditions, the system was tested at various airports to ensure adequate representation of landing geometry with reference to LLA coordinates. Successful landings indicated that the landing geometry definitions are in fact appropriate for use at any destination airport.

X-Plane Interface

Concentrating in aerial vehicles, X-Plane is used to verify/validate/refine controllers designed in MATLAB. Although there exist several simulators like Microsofts Flight Simulator and Flight Gear, XPlane provides extremely accurate flight models and allows for external communication as well as airfoil design. It is accurate enough to be used to train pilots [15]. Unlike the Microsoft Flight Simulator and Flight Gear, however, X-Plane also allows for input and output from external sources. As noted in [16], X-Plane provides future capabilities that unmanned aerial vehicles will need, including navigation markers, changing weather conditions and air traffic control communication.

A key aspect of controller verification / validation / refinement is the actual communication and interface between the X-Plane and MATLAB/ SIMULINK shown in Fig. 37.

X-Plane UDP Communication

X-Plane uses UDP communication to send and receive data packets; this allows for changes to be made to various values within X-Plane. The UDP protocol has advantages and disadvantages. UDP may be unreliable over a distant network connection because no error detection exists in the packets; on the other hand, UDP is extremely fast [17]. X-Plane is able to dump up to 50 frames per second across a local network; this has an important impact on controller functionality (and simulation) because they require sufficient update speed to operate correctly. X-Plane offers several parameters whose values may change,including control of the aircraft, failure introduction, etc.In order to select the appropriate data items to export to SIMULINK, X-Plane provides an easy to use checkbox interface.However, X-Plane is not open-source,hence, it will be necessary to be familiar with the UDP documentation [14].

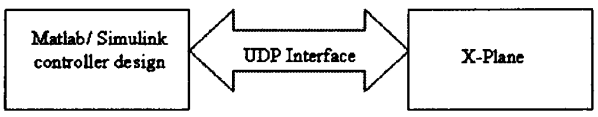

Fig.37 X-plane interface

The path followed by the aircraft after interfacing X-Plane with Simulink Via UDP are shown in Fig.38 and Fig.39. The aircraft landing sequence in X-Plane is shown in Fig.40-43.

Conclusion

The overall goal of this paper is the design and development of autonomous landing for an UAV and verification under simulation environment. Pitch angle and airspeed autopilots are required to make the successful landing and are designed appropriately. Pitch control is used to fly an UAV on a defined glide path and then through a flare maneuver to achieve touchdown 500 ft after the runway threshold. Separate controllers for glideslope control and flare control are designed. The glide path controller is shown to respond well to disturbances and the

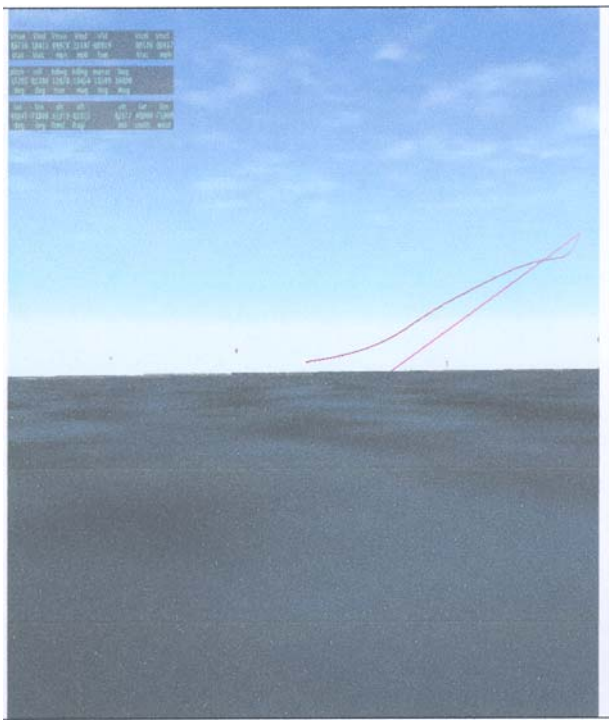

Fig.38 Desired and actual trajectory of UAV

flare controller is shown to meeting ideal conditions. The transition from glide slope tracking to the flare maneuver is initially performed using a threshold switch based on the altitude. The switching process led to the requirement of large control effort, driving the actuators into saturation. This causes considerable oscillations in pitch angle. To avoid this, a blending function has been designed, which provides a weighted combination of glide slope and flare controller commands during the transition between glide slope to flare. From the simulation results, it is inferred

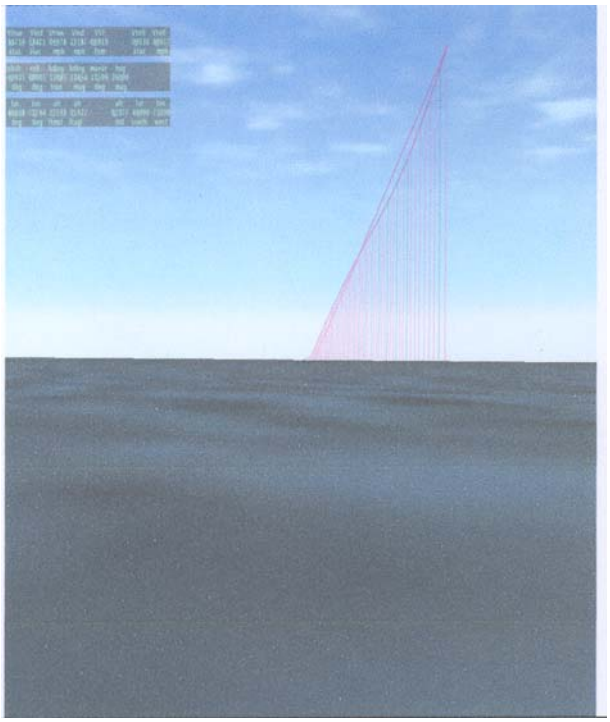

Fig.39 Desired and actual trajectory of UAV with waypoints

Fig.40 Aircraft aligning to the airport

Fig.41 Aircraft in glide path

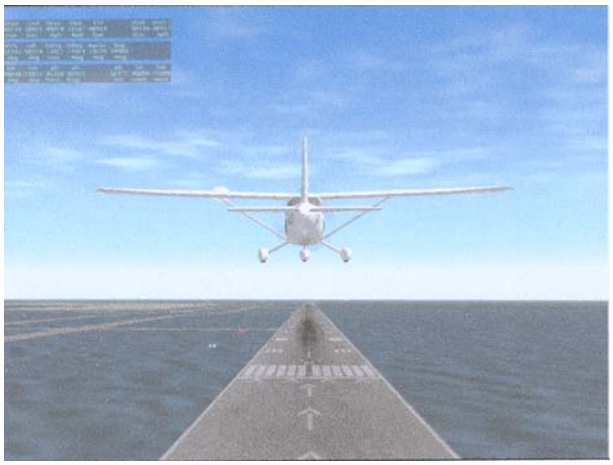

Fig.42 Aircraft in flare path

Fig.43 Aircraft towards touch down

that the blending of signals during transition from glide slope to flare solves the problem of instability and extreme oscillations. Successful landings are demonstrated at simulated airports around the world showing the versatility of the landing controller as it uses solely the destination airport for landing geometry and command reference. Although Lateral control is not addressed in this paper, it is indeed a challenge for the designer. It has been proposed to design lateral control laws at a future stage and integrate the system such as to obtain full real time simulation capability. The effect of sensor errors and transmission delays are the other problems which the designer faces. At present these effects have been assumed to be negligible and it might be worthy attempting it in future.

Acknowledgments

The authors would like to express their sincere thanks and gratitude to Dr. P. A. Janakiraman, Professor, Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, Shri P.S. Krishnan, Director and also Shri KV Srinivasan, Scientist G, Aeronautical Development Establishment, Bangalore for their inspiration and encouragement for successful completion of this work.

References

1. Blakelock J. H., "Automatic Control of Aircraft and Missiles", John Wiley Sons, New York, USA, 1990.

2. Wang Yongsheng., Li Xiangpeng and Huang Yong., "Navigation System of Pilotless Aircraft Via GPS", IEEE Aerospace and Electronic SystemsMagazine,Vol.11, No. 8, August 1996, pp.16-20.

3. Donald McLean., "Automatic Flight Control Systems", Prentice-Hall Publications, 1990.

4. TaoYong and ShenYongzhang., "Guidance and Control for Automatic Landing of UAV", J. Transactions of Nanjing, University of Aeronautics and Astronautics, 18 (2): 229-235, 2001.

5. Y. Li, Sundararajan, N., Saratchandran, P. and Wang, Z., "Robust Neuro-Hà, Controller Design for Aircraft Auto-landing", IEEE Trans. Aerospace Electronics. Syst., 40 (1), 2004, pp.158-167.

6. Charles C Jorgensen and Schley, C., "A Neural Network Baseline Problem for Control of Aircraft Flare and Touchdown", edited by W.Thomas Miller lll, Richard S. Sutton and Paul J.Werbos editors appeared in the "Neural Networks for Control", pp.403-425.

7. http://www.earth.google.com/.

8. Senthil Kumar, K, Sudhir Reddy and Shanmugam, J., "Design and Simulation of Blending Function for Landing Phase of an Unmanned Air Vehicle", Defence Science Journal, Vol.58, No. 3, 2008, pp.315-326.

9. Jan Roskam., "Airplane Flight Dynamic and Automatic Flight Controls", Part I and Part II, Roskam Aviation and Engineering Corporation, Kansas.

10. Senthil Kumar, K., Jagadeesh Balaji, P. and Shanmugam, J., "Development and Tuning of a Nonlinear Six DOF Model and Controllers for a Large UAV", International Journal of Aerospace Sciences and Technologies, Vol.59, No. 3, August 2007, pp.185-195.

11. Stevens, B.L. and Frank L Lewis., "Aircraft Control and Simulation", A Wiley-Inderscience Publication, 1992.

12. Control System Toolbox User's Manual, The Math Works Inc., 2005.

13. Simulink User's Manual and User's Guide, The Math Works Inc., 2004.

14. Meyer, A., X-Plane UDP Reference Manual, 2005.

15. Kreider, L., http://www.flightmotion.com/docs/faa_approval.htm., FAA Approval Document.

16. Walker, I. M. et al., "Simulation for the Next Generation of Civilian Airspace Integrated UAV Platforms", Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, Rhode Island, 2006.

17. Liu, P., Meng, M., Ye, X. and Gu, J., "An UDP-based Protocol for Internet Robots", Proceedings of the 4th World Congress Intelligent Control and Automation, China, June 2002.