A Robust Longitudinal Landing Controller to Datalink Time Delay

Sanghyo Lee*, Ihnseok Rhee**, Changdon Kee***, and Hueonjoon Koo****

* School of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Korea
(Tel : +82-2-880-8918; E-mail: ryanlee@snu.ac.kr)
** Control System Engineering, Korea University of Technology and Education, Chunan, Korea
(Tel : +81-41-560-1145; E-mail: rhee@kut.ac.kr)
*** School of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Korea
(Tel : +82-2-880-1912; E-mail: kee@snu.ac.kr)
**** Agency for Defense Delveolopment, Daejeon, Korea
(Tel : +82-42-821-3361; E-mail: junokoo@unitel.co.kr)

Abstract: This paper deals with designing a ground-based longitudinal landing controller which is robust to datalink time delays. Time delays occur because forward velocity measurements are downlinked and the controller output commands are uplinked. An H_∞ controller was designed by using the input/output decomposition where time delay is modeled as a first-order system with Pade approximation. Linear simulations show that the system tracks well the predefined path and is robust to the variation of time delay.

Keywords: Longitudinal Landing Controller, Datalink Time Delay, H_∞ , Flare

1. INTRODUCTION

This paper is aiming to design a ground-based longitudinal landing controller, which is robust to time delays of data link caused by a radar landing system. A radar landing system is chosen because automatic landing of an unmanned aerial vehicle (UAV) requires position information with high accuracy and it is needed to deploy as fast as possible in a harsh field, coping with interruption. However, this configuration, where positions are measured at the ground, has up and down data link where time delays occur. To reduce the period of development and verification, it is also required to use the existing autopilot as an inner-loop of overall landing controller. Therefore, we design an outer-loop command generator, using an H_∞ controller, where uplink and downlink time delays are modeled as a first-order system with Pade approximation. The inputs are forward velocity downlinked and altitude from a radar; the outputs, uplinked to the onboard autopilot, are pitch and throttle commands.

An H_∞ controller robust to parameter uncertainty is described in section 2. and then controller design requirements to develop an automatic landing system of UAVs are dealt with in section 3. Next, section 4 discusses construction of overall plant model for controller design and section 5 considers generation of forward velocity and altitude commands. Finally, linear simulations are performed in section 6. The results show that the system tracks well the predefined landing path and is robust to the variation of time delay.

2. H_∞ CONTROLLER FOR PARAMETER UNCERTAINTY

An H_∞ controller synthesis robust to parameter uncertainty is described here, summarizing the results of reference [1]. Parameter uncertainties can be equivalently represented as an internal feedback loop with the input/output decomposition

Consider a time-invariant linear system described by

$$\dot{x} = (A_0 + \Delta A)x + (B_0 + \Delta B)u\,, \tag{1}$$

$$z = (H_0 + \Delta H)x \tag{2}$$

where x, u, and z are the state vector, the input vector, and the measurement vector, respectively; A_0, B_0, and H_0 are the nominal system, input, and measurement matrices with (A_0, B_0) stabilizable and (H_0, A_0) dectectable; and ΔA, ΔB, and ΔH are perturbed matrices due to parameter variations.

The perturbations ΔA, ΔB, and ΔH can be decomposed as [1][2]

$$\Delta A = DL_a(\varepsilon)E, \quad \Delta B = FL_b(\varepsilon)G, \quad \Delta H = YL_h(\varepsilon)Z \tag{3}$$

where the matrices $L_a(\varepsilon)$, $L_b(\varepsilon)$, and $L_h(\varepsilon)$ are functions of the unknown parameter variation vector ε and the other matrices are known constant matrices.

By using the input/output decomposition in Eq. (3), the perturbed system in Eqs. (1) and (2) can be rewritten as

$$\dot{x} = A_0 x + B_0 u + \begin{bmatrix} D & F \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{4}$$

$$z = H_0 x + Yv \tag{5}$$

$$y = \begin{bmatrix} E \\ 0 \\ Z \end{bmatrix} x + \begin{bmatrix} 0 \\ G \\ 0 \end{bmatrix} u \tag{6}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ v \end{bmatrix} = L(\varepsilon)y = \begin{bmatrix} L_a(\varepsilon) & 0 & 0 \\ 0 & L_b(\varepsilon) & 0 \\ 0 & 0 & L_h(\varepsilon) \end{bmatrix} y \tag{7}$$

where, as shown in the Fig. 1, the uncertainty of the perturbed system is represented as a fictitious internal feedback loop caused by parameter variations.

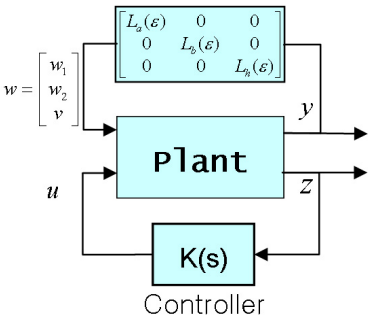

Fig. 1 The perturbed system with parameter variations

It follows, by small gain theorem [3], that controllers satisfying

$$\left\| T_{yw}(s) \right\|_\infty < \frac{1}{\gamma} \tag{8}$$

stabilize the closed-loop system under all the uncertainties of ε such that $\|L(\varepsilon)\|_\infty \le \gamma$. $T_{yw}(s)$ is the transfer function from w to y.

One of controllers that satisfy Eq. (8) is as follows [1][3]:

$$\dot{x}_c = A_c x_c + B_c z \tag{9}$$

$$u = C_c x_c \tag{10}$$

where

$$A_c = A_0 - B_0 R^{-1} B_0^T \Pi - M H_0^T V^{-1} H_0 + \gamma^2 W \Pi$$
$$B_c = M H_0^T V^{-1}, \; C_c = -R^{-1} B_0^T \Pi$$
$$M = (I - \gamma^2 P \Pi)^{-1}$$

if there exist $\mathbf{\Pi} \ge 0$, $P>0$, and $M>0$ satisfying the two AREs:

$$A^T \Pi + \Pi A - \Pi (B R^{-1} B^T - \gamma^2 W) \Pi + Q = 0 \tag{11}$$

$$AP + PA^T - P(H^T V^{-1} H - \gamma^2 Q) P + W = 0 \tag{12}$$

In Eqs. (11) and (12), the weighting matrices is given by

$$Q = \rho \bar{Q} + (E^T E + Z^T Z)$$
$$R = \rho \bar{R} + G^T G$$
$$W = \rho \bar{W} + (D D^T + F F^T)$$
$$V = \rho \bar{V} + Y^T Y$$

where ρ is scalar; $\bar{Q}$, $\bar{R}$, $\bar{W}$, and $\bar{V}$ are weighting matrices set in terms of performance which can also satisfy such conditions as stabilizable (Q, A_0) and detectable (A_0, W) that solutions of AREs in Eqs. (11) and (12) exist.

The controller given by Eqs. (9) ~ (12) guarantees stability robustness to given parameter variations, but is very conservative. The design parameters ρ and γ are chosen so that the AREs in Eqs. (11) and (12) have a nonnegative definite solution and a positive definite solution, respectively. As the value of ρ increases, system performance improves, whereas the value of γ increases, stability robustness with respect to parameter variation improves.

3. CONTROLLER DESIGN REQUIREMENTS

Our purpose is to develop an automatic landing system of UAVs. To land an aircraft automatically, it is essential to measure position with high accuracy, especially altitude. Altitude information is directly connected to aircraft safety. Since the landing system should be used in harsh field and should be deployed quickly, a high-precision ground radar system is selected as a position sensor for automatic landing.

The requirements imposed to develop a landing controller are to use the existing autopilot if possible and to install the landing controller in the ground station so that the control algorithm in UAVs does not have to be modified. The existing autopilot has selectable switches which make different autopilot modes.

Pitch autopilot loop is chosen as an inner-loop of overall landing controller and throttle command is directly engaged to the plant. Fig. 2 shows an existing inner autopilot loop with longitudinal stability augmentation system(SAS).

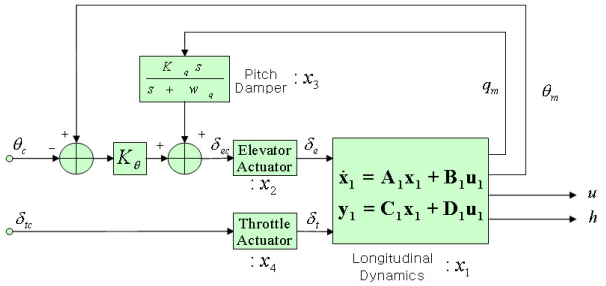

Fig. 2 The inner-loop Autopilot

Ground-based automatic landing controller with a radar landing system causes datalink time delay. During landing approach, aircraft velocity must be tightly controlled to prevent its stall, so forward velocity is downlinked. This causes downlink time delay. Commands are uplinked to the onboard autopilot, which also causes uplink time delay

The configuration of overall landing controller is represented as in Fig. 3. Ground landing controller is a kind of command generator and composes the outer-loop of overall landing controller. The inputs are reference command, forward velocity downlinked, and altitude from a radar; the outputs, uplinked to the onboard autopilot, are pitch and throttle commands.

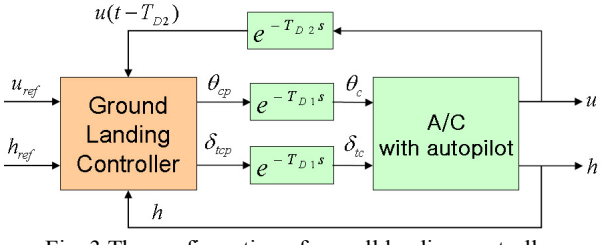

Fig. 3 The configuration of overall landing controller

There are a couple of design requirements that must be met to construct a landing controller. The aircraft must track the glide-slope without steady state errors and it must also track a given forward velocity without steady state errors. Before the touchdown, positive pitch angle is needed to nose up the aircraft , thus causing the main landing wheel to touch the ground first. The controller must be robust to some bound of uplink and downlink time delays. To meet these requirements, next section deals with the construction of plant model for controller design.

4. SYSTEM MODELING

4.1 Airplane Model for Landing

In an automatic landing problem, a controller has to be designed to follow given velocity and altitude commands. Velocity command is given by step input and glide slope is given by ramp input with respect to time.

Longitudinal dynamics of an aircraft can be given by

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = A \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + B \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} \tag{13}$$

and the altitude equation is given by

$$\dot{h} = C \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} \tag{14}$$

where u, α, q, θ, and h are forward velocity, angle of attack, pitch rate, pitch angle, and altitude, respectively; δ_e and δ_t are elevator and throttle inputs; and A, B, and C matrices are described in [5]. . Note that climb-rate $\dot{h}$ is not a function of h.

Augmenting Eq. (14) to Eq. (13) gives

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \\ h \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} \tag{15}$$

Eq. (13) has no eigenvalue at zero and is a sytem of type 0. Integrator of forward velocity u is necessary to follow step input of u. Eq. (15) has an eigenvalue at zero and thus is a system of type 1. Altitude state has zero steady state error response to step altitude input. However, altitude state also needs integrator since it has to follow the ramp input.

Let the velocity command and altitude command be $u_{cmd}(t)$ and $h_{cmd}(t)$, respectively. Then, velocity and altitude errors are defined as follows:

$$e_u = u - u_{cmd}(t) \tag{16}$$

$$e_h = h - h_{cmd}(t) \tag{17}$$

If velocity and altitude commands are step and ramp inputs, respectively, Eq. (15) can be rewritten as

$$\begin{bmatrix} \dot{e}_u \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{e}_h \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} e_u \\ \alpha \\ q \\ \theta \\ e_h \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} + \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{cmd} \\ 0 \\ 0 \\ 0 \\ \dot{h}_{cmd} \end{bmatrix} \tag{18}$$

where u_{cmd} and $\dot{h}_{cmd}$ are constant

To eliminate u_{cmd} and $\dot{h}_{cmd}$, we differentiate both sides of Eq. (18) and obtain

$$\begin{bmatrix} \ddot{e}_u \\ \ddot{\alpha} \\ \ddot{q} \\ \ddot{\theta} \\ \ddot{e}_h \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} \dot{e}_u \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{e}_h \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \begin{bmatrix} \dot{\delta}_e \\ \dot{\delta}_t \end{bmatrix} \tag{19}$$

Note that Eqs. (15) and (19) have the same coefficient matrices but the states and inputs of Eq. (19) are differentiated with respect to time.

4.2 Modeling of Time Delay

Time delay is modeled as a first-order system with Pade approximation, which gives

$$e^{-T_D s} \cong \frac{1 - \frac{T_D}{2} s}{1 + \frac{T_D}{2} s} \tag{20}$$

and the state-space representation of Eq. (20) can be written as

$$\dot{x}_d = A_d x_d + B_d u_d \tag{21}$$

$$y_d = C_d x_d + D_d u_d \tag{22}$$

where

$$A_d = -\frac{2}{T_D}, B_d = 1, C_d = \frac{4}{T_D}, \text{and } D_d = -1 \tag{23}$$

To consider the variation of time delay, time delay T_D is represented as the sum of nominal time delay T and perturbed time delay Δ, thus yielding

$$A_d = -\frac{2}{T_D} = -\frac{2}{T+\Delta} = -\frac{2}{T} + \frac{2\Delta}{T(T+\Delta)} = -\frac{2}{T} + \frac{2}{T}\tilde{\Delta} = A_{d0} + \Delta A_d \tag{24}$$

$$C_d = \frac{4}{T_D} = \frac{4}{T+\Delta} = \frac{4}{T} - \frac{4\Delta}{T(T+\Delta)} = \frac{4}{T} - \frac{4}{T}\tilde{\Delta} = C_{d0} + \Delta C_d \tag{25}$$

where

$$\tilde{\Delta} = \frac{\Delta}{T+\Delta}, A_{d0} = -\frac{2}{T}, \Delta A_d = \frac{2}{T}\Delta, C_{d0} = \frac{4}{T}, \text{and } \Delta C_d = -\frac{4}{T}\Delta$$

4.3 Plant Model For controller design

Adding e_u and e_h to the state vector of Eq. (19) yields

$$\dot{\tilde{x}}_1 = \tilde{A}_1 \tilde{x}_1 + \tilde{B}_1 \tilde{u}_1 \tag{26}$$

where

$$\tilde{x}_1 = \begin{bmatrix} e_u & e_h & \dot{e}_u & \dot{\alpha} & \dot{q} & \dot{\theta} & \dot{e}_h \end{bmatrix}^T \text{ and } \tilde{u}_1 = \begin{bmatrix} \dot{\delta}_e & \dot{\delta}_t \end{bmatrix}^T$$

Fig. 4 shows the airplane model with autopilot which is described by

$$\dot{x}_a = A_a x_a + B_a u_a \tag{27}$$

$$y_a = C_a x_a \tag{28}$$

where $x_a = \begin{bmatrix} \tilde{x}_1^T & \tilde{x}_2^T & \tilde{x}_3^T & \tilde{x}_4^T \end{bmatrix}^T$, $u_a = \begin{bmatrix} \dot{\theta}_c & \dot{\delta}_{tc} \end{bmatrix}^T$, and $y_a = \begin{bmatrix} e_u & e_h \end{bmatrix}^T$. $\tilde{x}_2$ and $\tilde{x}_4$ represent states of actuators and $\tilde{x}_3$ is the state of pitch damper.

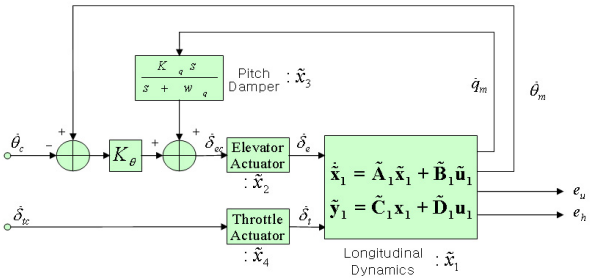

Fig. 4 Airplane model with autopilot for controller design

Uplink and downlink time delay models are combined with Eqs. (27)~(28) as shown in Fig. 5. Overall plant model for constructing a controller is given by

$$\dot{x} = Ax + Bu \tag{29}$$

$$z = Hx \tag{30}$$

where $x = \begin{bmatrix} x_a^T & x_{d1} & x_{d2} & x_{d3} \end{bmatrix}^T$, $u = \begin{bmatrix} \dot{\theta}_{cp} & \dot{\delta}_{tcp} \end{bmatrix}^T$, and $z = \begin{bmatrix} e_u(t - T_{D2}) & e_h \end{bmatrix}^T$.

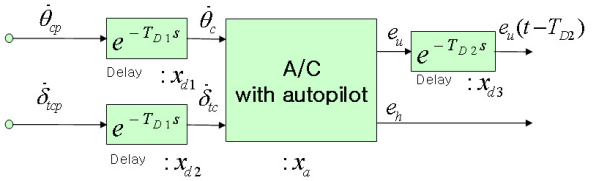

Fig. 5 Overall plant model for controller design

The perturbations due to time delay can be decomposed as

$$\Delta A = DL_a(\varepsilon)E, \quad \Delta B = 0, \quad \Delta H = YL_h(\varepsilon)Z \tag{31}$$

where

$$D = \begin{bmatrix} -B_a(:,1)\frac{4}{T_1} & -B_a(:,2)\frac{4}{T_1} & 0_{12\times 1} \\ \frac{2}{T_1} & 0 & 0 \\ 0 & \frac{2}{T_1} & 0 \\ 0 & 0 & \frac{2}{T_2} \end{bmatrix}, L_a(\varepsilon) = \begin{bmatrix} \tilde{\Delta}_1 & 0 & 0 \\ 0 & \tilde{\Delta}_1 & 0 \\ 0 & 0 & \tilde{\Delta}_2 \end{bmatrix}, E = \begin{bmatrix} 0_{3\times 12} & I_{3\times 3} \end{bmatrix},$$

$$Y = \begin{bmatrix} -\frac{4}{T_2} \\ 0 \end{bmatrix}, L_h(\varepsilon) = \tilde{\Delta}_2, \text{ and } Z = \begin{bmatrix} 0_{1\times 14} & 1 \end{bmatrix}.$$

and $B_a(:,1)$ and $B_a(:,2)$ represent the first and second columns of B_a, respectively.

A H_∞ controller discussed in section 2 can be designed for the plant in Eqs. (29) and (30). The resulting controller outputs are $\dot{\theta}_{cp}$ and $\dot{\delta}_{tcp}$. Since the inputs of the autopilot are θ_{cp} and δ_{tcp}, the controller outputs should be integrated before they are uplinked to the autopilot. Fig. 6 shows the structure of ground landing controller where two integrators are involved. As described in section 4.1, those integrators remove the steady state errors in velocity and altitude.

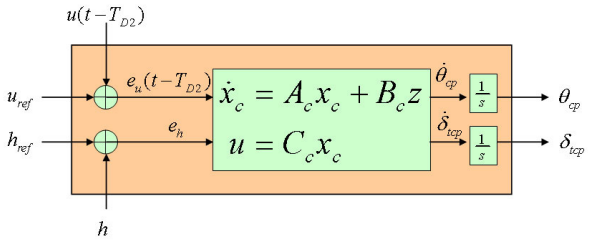

Fig. 6 Ground landing controller

5. COMMAND GENERATION

Forward velocity command is given by constant or step input. Velocity below trim value is required to nose up the aircraft before touch down. There is switching logic in altitude command, which consists of altitude hold, glide slope, and flare. First, constant altitude is generated and then ramp input command is activated when the aircraft passes the glide slope. Finally, exponential altitude command is generated in flare mode when the aircraft goes below designed flare altitude.

We used glide slope of 5 deg to shorten the length of landing approach, which is made possible because the landing velocity of UAV is relatively small, about 30m/s. Note that the important factor is the aircraft vertical velocity on the glide slope path. According to the simulation results, the designed controller cannot follow glide slope path of more than 7 deg because it also control the aircraft velocity which is increased due to gravity force as glide slope increases.

Reference altitude in flare path is generated exponentially as shown in Fig. 7. Flare path is tangential to the glide slope path. Design parameters are the forward horizontal landing distance, vertical velocity on the glide slope, and vertical velocity at the touchdown point. The relations between variables can be derived as follows:

$$h_0 = x_0 \tan\Gamma \tag{32}$$

$$x_t = x_0 \ln\frac{\dot{h}_t}{\dot{h}_g} \tag{33}$$

$$h_t = h_0 \ln\frac{\dot{h}_t}{\dot{h}_g} \tag{34}$$

where subscripts g and t denote glide slope and touchdown point. This flare path is suggested for the reference and tuning is required to achieve good landing performance in simulations and flight tests [4]. Note that true altitude at the beginning of flare path is h_0-h_t. It is helpful to represent the relations in Eqs. (32) ~ (34) in terms of time and horizontal velocity instead of landing distance.

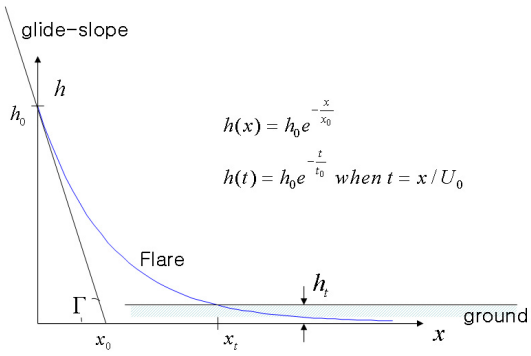

Fig. 7 Flare Path

6. SIMULATION

We performed linear simulations in order to verify the performance of the designed controller. The trim condition for steady level flight is at U_0 = 30.54 m/s and Θ_0 = 5.7 deg. Given nominal uplink time delay T_1 and downlink time delay T_2 are equally 0.15 sec. Different combinations of uplink and downlink time delay variations Δ_1 and Δ_2 were tested. Downlink time delay had less effect than uplink time delay. Downlink time delay is introduced due to downlinked forward velocity measurements and forward velocity, which is in the phugoid mode, changes slowly. Therefore, quantity of change of forward velocity for the downlink time delay is small. Performance is not degraded much up to down link time delay of 1 sec. On the other hand, the influence of uplink time delay is considerable. As uplink time delay increases, the system shows more oscillation and becomes unstable eventually. Landing performance was acceptable at the perturbed uplink time delay of 0.3 sec.

As mentioned in section 5, glide slope of 5 deg was used. Forward velocity commands are zero during the altitude hold and glide slope hold; however, in the flare mode, forward velocity command of -2 m/s is set to nose up the aircraft before touchdown.

Figs. 8 ~ 14 show some results of simulation. Figs. 8 ~ 11 are the results of no time delay variation. As shown in Fig. 8, the landing path is well tracked although responses are slow and overshoots are a little large when the altitude command mode changes. Note that integrators works well. It is shown in Fig. 9 that forward velocity converges to -2m/s and pitch angle is positive in flare mode before touchdown. Total pitch angle Θ at the touchdown point is about 7 deg considering the trim condition. Fig. 10 shows elevator and throttle inputs and Fig. 11 shows vertical velocity. Note that vertical velocity is maintained to about -2.6 m/s and then reduces to -0.2 m/s. Therefore, the aircraft can touch down safely.

Figs. 12 ~ 14 show the simulation results where both uplink and downlink time delay perturbations are 0.3 sec. More oscillatory and slow response is observed, but the vertical velocity is acceptable at touchdown.

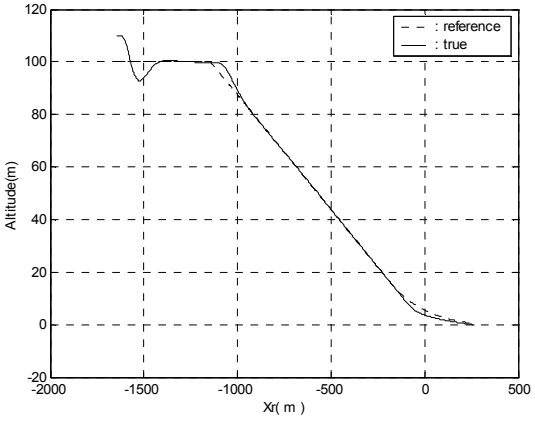

Fig. 8 Longitudinal landing trajectory(Δ_1=Δ_2=0)

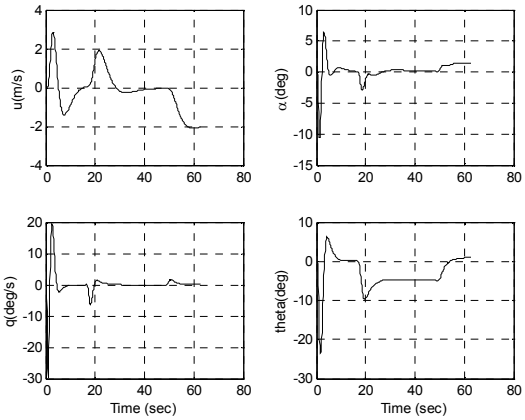

Fig. 9 Perturbed states(Δ_1=Δ_2=0)

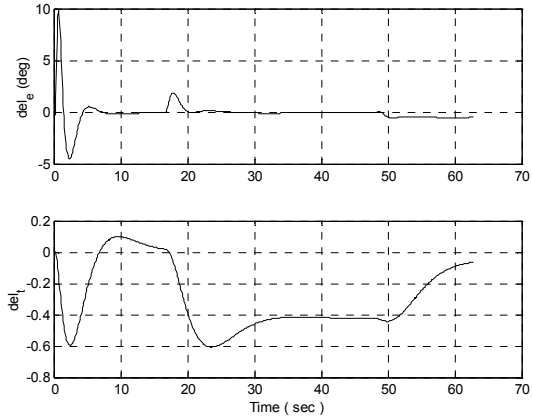

Fig. 10 Control inputs(Δ_1=Δ_2=0)

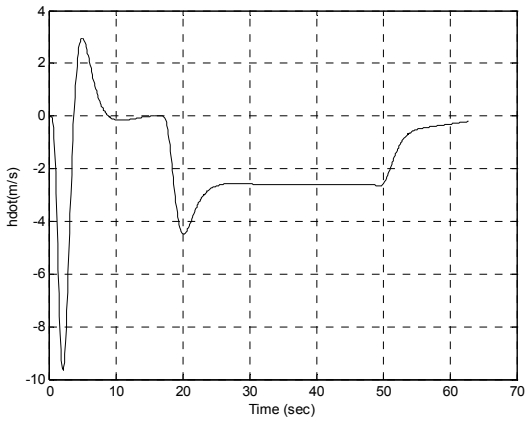

Fig. 11 vertical velocity(Δ_1=Δ_2=0)

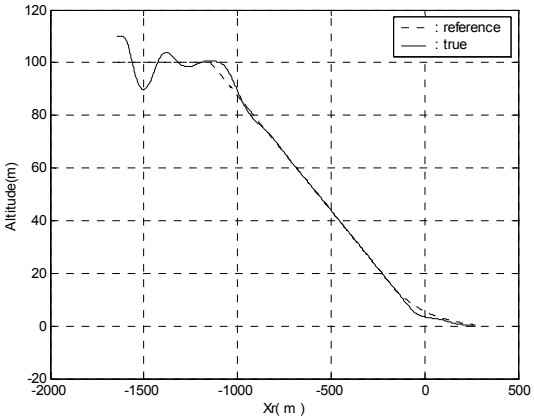

Fig. 12 Longitudinal landing trajectory($\Delta_1=\Delta_2=0.3$sec)

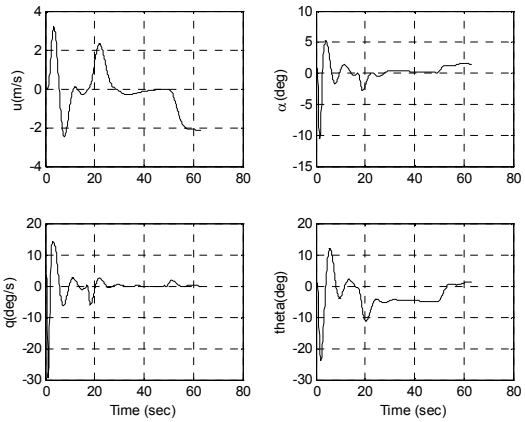

Fig. 13 Perturbed states($\Delta_1=\Delta_2=0.3$sec)

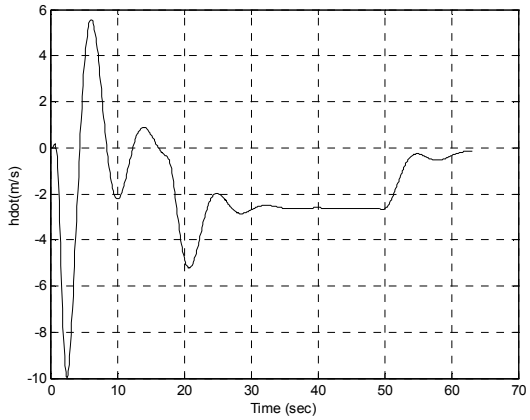

Fig. 14 vertical velocity($\Delta_1=\Delta_2=0.3$sec)

5. CONCLUSIONS

During landing approach, it is important to follow the glide-slope path and the given forward velocity command with no steady state errors. Hence, integrators need to be added to the states of forward velocity and altitude because altitude command is given by ramp input and forward velocity command is given by step input. A new approach was introduced to include integrators. The states are differentiated once with respect to time, and error states of forward velocity and altitude are augmented. Then, controller outputs are integrated before they are applied. Time delay is modeled as a first-order system with Pade approximation and is decomposed into nominal and perturbed values. We designed a H_∞ controller using the input/output decomposition where the uncertainty of the perturbed plant is represented as a fictitious internal feedback loop caused by time delay variations.

In the glide slope mode, glide slope of 5 degree was used to shorten the landing path and forward velocity command was set to zero. In the flare mode, reference altitude was generated exponentially to decrease descending velocity and forward velocity command of -2 m/s was set to nose up the aircraft before touchdown, thus yielding positive pitch angle. Linear simulation results show that the predefined landing path is well tracked although responses are slow and overshoots are a little large when the altitude command mode changes. Downlink time delay caused by forward velocity measurements had less effect than uplink time delay caused by controller output commands. Landing performance was acceptable at the perturbed time delay of 0.3 sec, where more oscillatory and slow response was observed, but the descending rate was small enough for the aircraft to touch down safely.

ACKNOWLEDGMENTS

This paper is sponsored by Agency for Defense Development.

REFERENCES

[1] Ihnseok Rhee and J. L. Speyer, "Application of a Game Theoretic Controller to a Benchmark Problem," Journal of Guidance, Control, and Dynamics, Vol. 15, No. 5, pp. 1076-1081, 1992.

[2] M. Tahk, and J. L. Speyer, "Modeling of Parameter Variations and Asymtotic LQG Synthesis," IEEE Transactions on Automatic Control, Vol. AC-32, No. 9, pp. 793-801, 1987.

[3] K. Zhow, J. Doyle and K. Glover, Robust and Optimal Control, Prentice Hall, 1996

[4] A. E. Bryson Jr., Control of Spacecraft and Aircraft, Hemisphere, New York, 1995.

[5] H. Koo, “The Comparison of Analytical Results with Flight Test of Algorithm Performance for UAV’s Autopilot and Operation Logic,” ASDC-501-971207, Agency for Defense Development, 1997.