AUTOMATIC LANDING CONTROL SYSTEM

RASHMI KOUSHIK & K. ARULMOZHI

School of Aeronautical Sciences, Hindustan Institute of Technology and Science, Padur, Kelambakkam, Chennai, India

ABSTRACT

Approach and landing is the highest risk phase of flight, accounting for over 50 percent of all accidents at every level of aviation. Many types of accidents can happen during the approach and landing phase of flight.

Because of its inherent risk there have been many international efforts addressing the approach and landing phase of flight. These efforts have primarily addressed the highest risk areas such as CFIT and runway excursions. They have addressed all aspects of approach and landing accidents, and in addition they also addressed all the responsible parties involved with reducing the risk of approach and landing accidents. These parties include the aircraft manufacturers, aircraft operators, aircrews, air traffic management, regulators, and airports. Many interventions have been created to assist in reducing the risk of accidents in the approach and landing phase of flight.

Some instruments can be used to land the aircraft in limited visibility by positioning the aircraft relative to a reference trajectory. However, automatic controls are to be applied when the conditions are at the worst.

The objective of this project is to devise an automatic landing control system using Model- Following control and H-Infinity output feedback control and test it on X-Plane flight simulator.

KEYWORDS: Landing System, Controller & H-Infinity Output Feedback Control

Received: Jun 08, 2020; **Accepted:** Jun 28, 2020; **Published:** Aug 11, 2020; **Paper Id.:** IJMPERDJUN2020726

INTRODUCTION

Model-Following Controllers

Model-following controllers are designed to eradicate differences between the process variable and the set point by forcing the process variable to reach the set point along a specified trajectory. The suitable trajectory specified by the user by creating a mathematical model that represents an idealized process that would provide a more ideal response to a set-point change if replaced by the original process. The controller then calculates the output and not the actual process variable and attempts to drive the model's output in the direction of the set-point along the desired trajectory.

This algorithm, known as "model following," can be accomplished with two proportional-integral-derivative (PID) controllers. In the top loop, known as the "model loop," the "model controller" applies its corrective efforts or "model control signal" to the mathematical model as if it were a real process. The model's output is fed back and subtracted from the set-point to generate the error signal that feeds into the model controller. [8]

The model control signal is tried with the original process along with some increment in the form of a "correcting signal" produced by the "correcting loop." The error signal for this loop is the distinction between the model's output and the actual process variable. That difference is input to the "correcting controller" to generate the correcting signal which is added to the model control signal. That sum serves as the control effort applied to the

actual process. [8]

The model controller is designed to achieve whatever trajectory the user wants the model's output to follow in response to a setpoint change. This is a straightforward exercise since the behavior of the idealized process is already known, having been defined by the user. Any number of loop-tuning rules could be used to translate the gains and time constants of the idealized process model into tuning parameters for the model controller.

Tuning the correcting controller can be trickier, especially if the gains and time constants of the real process are unknown or time-varying. The correcting loop is also subject to real-world load disturbances plus artificial disturbances caused by the model control signal.

Major advantages of model following control are Model-following controllers have proven useful for applications where robustness is required. They tend to be less sensitive to behavior variations of the controlled process than traditional single-loop control strategies. [8]

H_∞ (I.E. "H-Infinity") Methods

H_∞ (i.e. "H-infinity") methods are used in control engineering to produce controllers to attain amplified stabilization with assured execution. We have used H-Infinity output feedback controller to calculate feedback gains for the available state variables since not all of the state variables will be available in actual flight. Determining initial stabilizing gain is not required in this technique [2].

H_∞ techniques can be used to minimize the closed loop impact of a perturbation: depending on the problem formulation, the impact will either be measured in terms of stabilization or performance. Simultaneously optimizing robust performance and robust stabilization is difficult.

Robust H-infinity control can give a perfect control to linear systems and high robustness to stabilize in unfavorable operating conditions such as parameter change, high disturbance environment actuator saturation, and model uncertainty. [15]

X-Plane Simulator

X-Plane is software which is certified by Federal Aviation Administration (FAA). It is used to instruct pilots so as to achieve flight as in authentic situations. X-Plane is an engineering tool that can predict the performance and handling of any aircraft [9]. Recently X-Plane has been used by researchers to design and test some projects on it to have an idea how the projects work in real life. X-Plane includes many commercial airlines and military based aircrafts in default. Moreover in plane-maker mode researcher can design a new aircraft and fly it on the simulator. The main advantage of this flight simulator is that X-Plane allows users to receive and send data via User Datagram Protocol (UDP). In other words, flight parameters can be received from it and control surface commands can be sent to the program to deflect aircraft control surfaces. Furthermore, in X-Plane there is an option which can change the weather situations. For instance, wind speed and wind direction can be set in order to give some disturbance to the aircraft so that the robustness of the controller can be evaluated.

AIRCRAFTDYNAMICS

There is a wide variety of aircrafts available in X-Plane. One of them is B747-400. Its functioning is widely researched and studied. Considering these facts, B747-400 aircraft is chosen to be used in this paper.

The compact form of the state space equation is written in (2.1);

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$$

$$\mathbf{x} = [V_T \ \alpha \ \theta \ q \ h \ h_{ref}]' \quad (2.1)$$

B747-400 aircraft is linearized by using the small disturbance theory [3]. Landing configuration for Boeing 747 is at sea level, power approach, 30 degrees flaps down, gear down and that condition is selected as the equilibrium point [5]. The state space equation for B747 longitudinal dynamics for landing flare are shown in (2.2).

$$\begin{bmatrix} \dot{V}_T \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \\ \dot{h} \\ \dot{h}_{ref} \end{bmatrix} = \begin{bmatrix} \frac{X_u}{U_0 - Z_u} & \frac{X_\alpha}{U_0 - Z_\alpha} & \frac{0}{U_0 - Z_\alpha} & \frac{-g \cos \theta_0}{U_0 - Z_\alpha} & 0 & 0 \\ \frac{Z_u}{U_0 - Z_u} & \frac{Z_\alpha}{U_0 - Z_\alpha} & \frac{Z_q + U_0}{U_0 - Z_\alpha} & \frac{-g \sin \theta_0}{U_0 - Z_\alpha} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ M_u + Z_u T & M_\alpha + Z_\alpha T & M_q + (Z_q + U_0) T & -g(\sin \theta_0) T & 0 & 0 \\ 0 & -U_0 & U_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_T \\ \alpha \\ \theta \\ q \\ h \\ h_{ref} \end{bmatrix} + \dots$$

$$\begin{bmatrix} \frac{X_{\delta_1}}{Z_{\delta_1} T} & \frac{X_{\delta_2}}{Z_{\delta_2} T} \\ 0 & 0 \\ M_{\delta_1} + Z_{\delta_1} T & M_{\delta_2} + Z_{\delta_2} T \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix}, (T = M/(U - Z)) \quad (2.2)$$

The variables in (2.1) are; the deviation from trim velocity: V_T , angle of attack: α , pitch rate: q , pitch attitude angle: θ , altitude: h , reference altitude h_{ref} , throttle commanded: δ_1 , elevator deflection: δ_2 . Names and numerical values of the aircraft dynamics parameters taken from [5] are shown in Table 1.

Table 1

SYMBOL	Parameters used in Aircraft Dynamics		
	Name of Parameters	Unit	Glide-Slope and Flare
X_u	Forward acceleration per unit change in speed	1/sec.	-0.0212
Z_u	Vertical acceleration per unit change in speed	1/sec.	-0.2445
M_u	Pitch angular acceleration per unit change in speed	1/sec.	2.0043e-05
X_α	Forward acceleration per unit angle of attack	rad./sec. ²	13.0169
Z_α	Vertical acceleration per unit angle of attack	rad./sec. ²	-167.8312
M_α	Pitch angular per unit angle of attack	rad./sec. ²	-0.5400
X_q	Forward acceleration per unit pitch rate	((ft./sec. ²)/rad./sec.)	0

Z_q	Vertical acceleration per unit pitch rate	((ft./sec. ²)/rad./sec)	-7.6697
M_q	Pitch angular acceleration per unit pitch rate	1/sec.	-0.4377
$Z_{\dot{\alpha}}$	Vertical acceleration per unit change of angle of attack	((ft./sec. ²)/rad./sec)	9.5161
$M_{\dot{\alpha}}$	Pitch angular acceleration per unit change of angle of attack	1/sec.	-0.0673
X_{δ_e}	Forward acceleration per unit elevator angle	((ft./sec. ²)/rad.)	0.9710
Z_{δ_e}	Vertical acceleration per unit elevator angle	((ft./sec. ²)/rad.)	-9.7300
M_{δ_e}	Pitch angular acceleration per unit elevator angle	1/sec. ²	-0.5740
X_{δ_t}	Forward acceleration per unit throttle command	(1/slugs)	5.700e-05
Z_{δ_t}	Vertical acceleration per unit throttle command	(1/slugs)	-2.4900e-06
M_{δ_t}	Pitch angular acceleration per unit throttle command	(1/slugs ft. ²)	3.1000e-07
U_0	Referenced forward speed	(ft./sec.)	278
θ_0	Referenced pitch attitude speed	deg.	0

Landing Flare

The flare control system is based on the model which is described in [1]. Landing of an aircraft begins with the approach session and then aircraft follows the glide slope line until flare point altitude between 20 and 70 ft above the end of the runway [1]. To reduce the rate of descent, the flare part of the landing is initiated. The flare path geometry is shown in fig 1.

Figure 1. Landing flare path geometry [1].

The parameters in Fig. 1 are; initial flare altitude, referenced flight path angle: γ_0 , the distance from the glide-slope transmitter to touchdown: D . Flare trajectory as it is seen in Fig. 1 exponentially decays. The exponential model is written in (2.1.1) and (2.1.2).

$$h = h_0 e^{-t/\tau} \quad (2.1.1)$$

$$\dot{h} = -\frac{h}{\tau} \quad (2.1.2)$$

$$\dot{h}(0) = -\tau \dot{h}(0) = \tau V_T \sin \gamma_R \quad (2.1.3)$$

The parameters in (2.1.3); total aircraft velocity: V_T , time constant τ , altitude h . The horizontal component of the aircraft's velocity is $V_T \cos \gamma$. It is approximately equal to V_T since flight path angle is so small ($\cos \gamma \approx 1$). The total horizontal distance from initial flare point to touchdown is written in (2.1.4)

$$4\tau V_T = \left(\frac{h_0}{\tan \gamma} \right) + D = \tau V_T \cos \gamma + D \quad (2.1.4)$$

$$\tau = \frac{D}{3V_T} (\cos \gamma \approx 1) \quad (2.1.5)$$

If D is taken as 3646 (ft.) and $V_T=278$ (ft./sec.), $\gamma_R=-2.5$ deg and then τ becomes 4.372 sec. Thus by using (2.1.3), initial altitude for flare becomes 53 (ft.). This altitude is the switch altitude from glide-slope to flare phase.

DESIGN OF THE CONTROLLERS

The trajectory drawn by the reference model in flare part of the landing is followed by the aircraft using Model-Following control technique. The H-Infinity output feedback control is used to calculate the feedback gains used in flare control model.

The main aim of the model following control is to make the aircraft behave like a desired model. Regulation with model following control technique is used in designing flare path. In other words, plant states are driven to zero in regulator problem case. The structure of the model-following control method is shown in fig 2.

Figure 2: MFC Control Structure.

In this study, landing flare path is drawn by the referenced model and commanded altitude is created. The landing flare path trajectory decays exponentially. Commanded altitude differential equation is described in the exponential reference model which is explained in landing flare section and it is written in (3.1), (3.2) and (3.3).

$$\dot{h} = -\frac{h}{\tau} \quad (3.1)$$

$$\dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}r \quad (3.2)$$

$$\dot{h} = -0.18h + r; h_0 = 53(\text{ft}) \quad (3.3)$$

Time constant “ τ ” is defined and calculated as 5.5 sec. by using desired flare conditions in landing flare and reference input “ r ” is set constant value of zero.

3.1 H-Infinity Output Feedback Control

H-Infinity output feedback design technique for linear time invariant systems is used in many areas of control engineering such as flight control systems [2]. There are some algorithms to find the H-Infinity design solution using output-feedback. These are Algebraic Riccati Equation based iterative numerical algorithms. Most of the iterative algorithms for output feedback control design require initial stabilizing gains [8]. In this study, the algorithm which does not require an initial stabilizing gain is used to design output feedback control. The main equation used in algorithm as described in [8] is written in (3.1.1).

$$P_n A + A^T P_n + Q + \frac{1}{\gamma^2} P_n D D^T P_n - P_n B R^{-1} B^T P_n + L_n^T R^{-1} L_n = 0 \quad (3.1.1)$$

$$K_{n+1} = R^{-1} (B^T P_n + L_n) C^T (C C^T)^{-1} \quad (3.1.2)$$

$$L_{n+1} = R^{-1} K_{n+1} C - B^T P_n \quad (3.1.3)$$

The gain parameter used for calculating controller gains in flare control block diagram is set as 0.5 in order to obtain good tracking results. The parameters in (3.1.1) are; state matrices: A, B, C and D, state weighting matrix: Q, input weighting matrix: R, gain parameter: γ , parameter matrix used in H-Infinity gain calculation: L, algebraic riccati solution matrix: P. Algebraic riccati solution solver in Matlab program is well developed to implement in solver algorithm [8].

X-PLANE CONNECTION

X-Plane is a flight simulator written by Laminar Research and it can be installed on computers which have various operating systems such as Linux, Windows, or Mac OS. X-Plane is not a game, but an engineering tool that can be used to predict the flying qualities of fixed-wing or rotary-wing aircraft with a significant accuracy [9].

It is possible to read and write flight parameters in this simulator. X-Plane has 133 channels which represent data output from it. These channels include aircraft control surfaces, throttle, speeds, atmosphere etc.

It is possible to read and write flight parameters in this simulator. X-Plane has 133 channels which represent data output from it. These channels include aircraft control surfaces, throttle, speeds, atmosphere etc.

User Datagram Protocol (UDP) is used to communicate with X-Plane. UDP protocol is a non-guaranteed protocol that gives no assurance that data packets will arrive in order or at all. However, X-Plane can receive up to 99.9 data packets per second across local network. This is enough speed to simulate and synchronize X-Plane and connected program because sufficient data can be obtained from it by that speed. However data rates cannot exceed the sending speed of the simulator [7].

X-Plane data format is 32 bits precision floating point. The first four bytes of the packet shown on Fig. 3 represents the characters "DATA" which indicates this is a data package. The fifth byte is an internal code "I". The next four bytes represent the parameter label "L1, L2, L3, L4". The Next 5 sets of 4 bytes "B11, B12, B13 ..." represent the data

itself in single precision floating point. The first byte is the sign bit which tells whether the number is positive or negative [7].

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	38	39	40	41
“D”	“A”	“T”	“A”	I	L 1	L 2	L 3	L 4	B 11	B 12	B 13	B 14	B 21	B 22	B 23	B 24	B 81	B 82	B 83	B 84

Figure 3: Data Package [7].

In X-Plane there is a section which is called "Net Connections" in X-Plane to set the appropriate parameters on program. If the X-Plane and the other programs such as Matlab-Simulink are running in the same computer, local host IP address "127.0.0.1" and port number "49003" can be used for communication. Otherwise, IPv4 IP addresses can be used. In this study local host IP address is used to communicate with X-Plane. Moreover, X-Plane UDP Port block has port settings which have to be chosen carefully since these are the gates to communicate with the simulator. 49000 is the default port for X-Plane to receive signals.

Augmented state vector for flare control and measured output state vector are written in (4.1) and (4.2) respectively.

$$\mathbf{X}_{\text{aug}} = [V_T \quad \alpha \quad \theta \quad q \quad h \quad \delta_t \quad \delta_e \quad X_V \quad X_1 \quad \varepsilon \quad h_{\text{ref}}]^T \quad (4.1)$$

$$\mathbf{y} = [\mathbf{X}_v \quad \mathbf{X}_1 \quad \varepsilon \quad \mathbf{e} \quad \mathbf{q} \quad \theta \quad h_{\text{ref}} \quad \mathbf{e}_v]^T \quad (4.2)$$

The parameters in (4.1) and (4.2) are; state variables in blocks: x_v , x_d , x_l , tracking errors in channels: e_v and e_d , the deviation from trim velocity: v_T , angle of attack: α , pitch rate: q , pitch attitude angle: θ , throttle commanded: δ_t , elevator deflection: δ_e , altitude: h , reference altitude h_{ref} .

Controller gains for landing flare control are all calculated by H-Infinity output feedback method. Controller gains used in Fig. 4 and weighing matrices are shown in Table 2.

Table 2: Control Design Parameters

K	Controller Gains	Weighing Matrices
k1	1.4279e-12	$Q = \begin{pmatrix} 10^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^2 & 0 \end{pmatrix}$
k2	2.5727e-02	
k3	8.5925e-01	
k4	-7.7358e-01	
k5	-8.0444e+01	
k6	-4.3750e+01	
k7	-1.7231e+05	
k8	1.1057e+03	$R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Linear model for flare control results obtained in Simulink are shown in Fig. 5, Fig. 6 and Fig. 7.

The augmented structure for landing flare control are shown in Fig. 4.

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

$$\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

$$y = Cx + Du$$

Figure 4: Landing Flare Control.

Results shown in Fig. 5, Fig. 6 and Fig. 7 are obtained in Matlab-Simulink environment for flare part of the landing by using the linearized model of B747 aircraft. Initial conditions for flare model are taken from the final glide-slope flight states. As it is seen in Fig. 5 flight path angle is initially -2.5 degree and linear model tries to make it zero, which is expected, in almost 20sec. In Fig.6 aircraft flare altitude and desired flare altitude are shown. Aircraft tries to catch the desired path in 10 sec. and then finally it dies out with some error which is small enough. The elevator response in landing flare is shown in Fig. 7. The elevator makes some oscillations first 3 seconds and then it is tuned as expected.

Figure 5. Angle of attack, pitch angle and flight path angle.

Figure 6. Desired flare altitude and aircraft altitude.

Figure 7. Elevator deflections.

X-Plane flight simulation results are shown in Fig. 8, Fig. 9, Fig. 10, Fig. 11 and Fig. 12

Figure 8. Aircraft landing altitude.

Figure 9. Aircraft flare altitude.

Figure 10. Angle of attack.

Figure 11. Pitch angle.

Figure 12. Flight path angle.

X-Plane flight simulator and Matlab-Simulink environment run simultaneously for all landing phases and the results of the flare part of the landing are shown. In glide slope phase of the landing which starts at 1500 feet altitude, aircraft tries to follow the glide slope line so that it has almost -2.5 degree flight path angle just before the flare phase which is at 53 feet altitude. At the initial stage of the flare phase of the landing, aircraft has almost zero degree angle of attack and -2.5 degree pitch angle and then the flight path angle is tried to be almost zero in the flare phase. Controllers which can be seen in flare control structure shown in Fig. 4 makes aircraft follow the reference model. The X-Plane model aircraft touchdown happens just a few seconds before the reference model.

CONCLUSIONS

In this study, model following controller with H-Infinity output feedback control algorithm are used to control landing flare phase and the flight for landing only is simulated in X-Plane flight simulator. Model following control is used to force the aircraft to follow the reference flare model in flare phase of the landing. H-Infinity output feedback control is also used to obtain controller gains. The controllers designed are tested in X-Plane flight simulator which has a realistic flight experience. The linear model with controllers demonstrates better matching results in Matlab-Simulink environment compared to X-Plane flight simulator.

REFERENCES

1. B. L. Stevens, F. L. Lewis, "Aircraft Control and Simulation" John Wiley & Sons, Inc., 1992.
2. B. Etkin; L. D. Reid, «Dynamics of Flight Stability and Control» New York, Wiley, 1959.
3. J. Roskam, "Airplane Flight Dynamics and Automatic Flight Controls", Design, Analysis and Research Corporation (DARcorporation), 2001.
4. Pavanan, Velmurugan, and Lakshminarayanan Varadharajan. "Optimization of Design and Operating Parameters for Enhancing the Performances of PEMFC with Serpentine Flow Fields." *International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)* 7.4 (2017): 447-456.
5. R. B. Gezer, A. T. Kutay "Robust Model Following Control Design for Missile Roll Autopilot" 2014 UKACC International Conference on, Loughborough, 2014, pp. 7-12. doi:10.1109/CONTROL.2014.6915107.
6. Laminar Research. "X-Plane 10 Desktop Manual", 2016.
7. Kumar, Gurram Narendra Santosh, and A. Srinath. "An Ergonomical condition's of Pedestrians on Accelerating Moving Walkway: A People Mover System." *International Journal of Mechanical and Production Engineering Research and Development* 8 (2018): 1376-1381.

8. C. Notteboom, M. Hochstenbach, "Data Connection between MATLAB/Simulink & X-Plane using UDP.pdf". Retrieved from <http://forums.x-plane.org/index.php?/forums/>.
9. Ender Çetin, Ali Türker Kutay, "Automatic Landing Flare Control Design by Model-Following Control and Flight Test on X-Plane Flight Simulator" Retrieved from 7th International Conference on Mechanical and Aerospace Engineering 2016.
10. Kumar, J. Jagadesh, G. Diwakar, and V. V. Satyanarayana. "Corrosion fatigue of welded joints in marine and naval structures: a review." *International Journal of Mechanical and Production Engineering Research and Development* 7.6 (2017): 155-162.
11. "The basics of model-following control" from <https://www.controleng.com/articles/the-basics-of-model-following-control/> 2019
12. Ismaila Bayo Tijani et al. "Optimization of an extended H-infinity controller for unmanned helicopter control using Multiobjective Differential Evolution (MODE)" 2016
13. Mihai Lungu, Romulus Lungu "The Control of Airplane Landing in Longitudinal and Lateral-directional Planes by using the H-inf Control" 2013
14. Taufik, Akhmad, Shingo Okamoto, and Jae Hoon Lee. "Multi-Rotor Drone to Fly Autonomously along a River Using a Single-Lens Camera and Image Processing." *International Journal of Mechanical Engineering* 4.6 (2015): 39-49.
15. ZHANG Jianfeng, JIA Caijuan, "Automatic Landing controller Design and Simulation of Flying-Wing Unmanned Aerial Vehicle" 2013
16. Hiroki Shibasaki et al., "A Design Method of a Model-following Control System " 2015
17. Ziyang Zhen et al., "An Adaptive Control Scheme for Carrier Landing of UAV" 2018
18. Pierre Apkarian* and Dominikus Noll, "The H_∞ control problem is solved", 2006
19. "Chapter 5 H-Infinity Controller", retrieved from <https://shodhganga.inflibnet.ac.in/bitstream/10603/141569/13/13>