drones

MDPI

Article

Autonomous Landing of an UAV Using H_∞ Based Model Predictive Control

Zohaib Latif [1,*], Amir Shahzad [2], Aamer Iqbal Bhatti [1], James Ferris Whidborne [3] and Raza Samar [2]

[1] Department of Electrical Engineering, Capital University of Science & Technology, Islamabad 44000, Pakistan
[2] Centres of Excellence in Science and Applied Technologies, Islamabad 44000, Pakistan
[3] Centre for Aeronautics, Cranfield University, Cranfield MK43 0AL, UK
* Correspondence: engr.zlatif@gmail.com; Tel.: +92-322-5259197

Abstract: Possibly the most critical phase of an Unmanned Air Vehicle (UAV) flight is landing. To reduce the risk due to pilot error, autonomous landing systems can be used. Environmental disturbances such as wind shear can jeopardize safe landing, therefore a well-adjusted and robust control system is required to maintain the performance requirements during landing. The paper proposes a loop-shaping-based Model Predictive Control (MPC) approach for autonomous UAV landings. Instead of conventional MPC plant model augmentation, the input and output weights are designed in the frequency domain to meet the transient and steady-state performance requirements. Then, the $\mathcal{H}_\infty$ loop shaping design procedure is used to synthesize the state-feedback controller for the shaped plant. This linear state-feedback control law is then used to solve an inverse optimization problem to design the cost function matrices for MPC. The designed MPC inherits the small-signal characteristics of the H_∞ controller when constraints are inactive (i.e., perturbation around equilibrium points that keep the system within saturation limits). The $\mathcal{H}_\infty$ loop shaping synthesis results in an observer plus state feedback structure. This state estimator initializes the MPC problem at each time step. The control law is successfully evaluated in a non-linear simulation environment under moderate and severe wind downburst. It rejects unmeasured disturbances, has good transient performance, provides an excellent stability margin, and enforces input constraints.

Keywords: autonomous landing; H_∞ synthesis; MPC; UAV

check for updates

Citation: Latif, Z.; Shahzad, A.; Bhatti, A.I.; Whidborne, J.F.; Samar, R. Autonomous Landing of an UAV Using $\boldsymbol{H_\infty}$ Based Model Predictive Control. *Drones* **2022**, *6*, 416. https://doi.org/10.3390/drones6120416

Academic Editor: Xiwang Dong

Received: 14 November 2022
Accepted: 3 December 2022
Published: 15 December 2022

1. Introduction

Over the past few decades, Unmanned Air Vehicles (UAVs) have gained popularity in various applications, from military operations to public safety. Nowadays, UAVs can perform commercial tasks such as image and data acquisition in disaster areas, communication delays, traffic surveillance, map building, search and rescue, and so on. With the rise in popularity, the number of incidents involving UAVs has increased drastically due to inexperienced operators. Most UAVs have autonomous take-off and cruise but limited autonomous landing capabilities due to reliability issues and high risks [1]. In the autonomous landing phase, the UAV performs many sensitive and vital tasks in different environmental conditions. Therefore, the performance of UAVs must not fail in the presence of external disturbances (e.g., crosswinds, fog).

The landing task has two phases, glide and flare maneuver. In the glide maneuver, the UAV must descend along a predefined straight-line path in the longitudinal plane with a fixed negative flight path angle between -3° to -4°, toward the runway. Sometimes, the glide maneuverability is performed in two steps. Initially, a higher decent angle is taken, and a lower descent angle is attained in the latter step. When the UAV reaches around 25–30 m altitude, the flare maneuver is initiated. In this phase, the UAV is required to reduce the descent rate and follow a curved path. It is necessary to bring the flight path angle near zero for a smooth touchdown and the minimum impact on the landing gears. Figure 1 illustrates the typical landing maneuver with the glide and flare path indication.

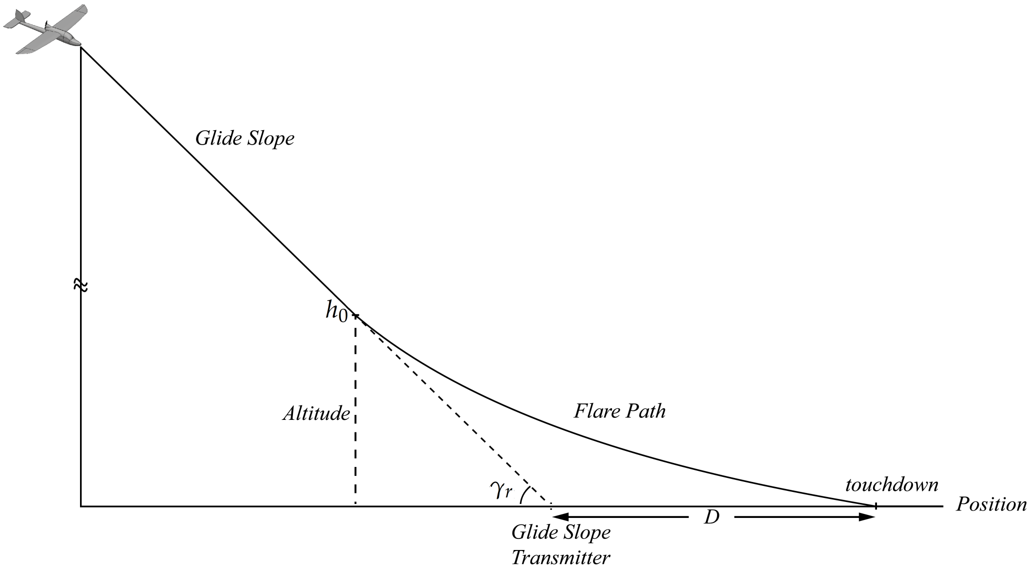


Figure 1. Reference Glide and Flare Path.

For most UAV operations, a human is used to pilot the landing. It is usually done to reduce the costs and complexity of the system and mitigate risks. Operational experience has shown that most UAV disasters are due to human errors [2]. Furthermore, it takes several hours of flight and significant financial investment to train a human pilot who can handle these phases efficiently and safely. There are also severe restrictions on a human pilot during flight operations; for example, he cannot land the UAV in adverse conditions such as thick fog and high crosswind. Thus, equipping the UAV with the ability to perform landing autonomously increases the system's complexity but can potentially render more versatile UAVs. It also can potentially reduce the long-term costs and risks involved in the landing. Moreover, external disturbances, such as wind gusts, shear, and downbursts, can be tackled more efficiently.

Classical control techniques are preferred for autonomous landing because of their reliability and simplicity. However, usually these techniques often do not satisfy the desired performance and robustness requirements. In recent years, with the developments in the hardware and computing capacity, the difficulties in implementing modern control techniques have been significantly reduced [3]. Therefore, the modern control techniques-based flight control system has become a mature research area. Many methodologies, from vision-based navigation to fuzzy logic control algorithms, have been implemented by researchers to achieve safe and smooth landings. In [4], a model following a technique based on Linear Quadratic Regulator (LQR) is given to track the glide and flare trajectory.

A flight algorithm for the autonomous landing of a UAV is proposed in [5]. Different control loops are used to address the specific task separately, e.g., angle of sideslip control or heading control. A multi-variable H_∞ controller is designed for flare maneuverability. At the top level, a non-linear guidance control law provides the 3D path following capability. A control law is proposed in [6] for Autonomous Carrier Landing (ACL) system. The non-linear landing model is transformed into a poly-topic model, and the arresting and ground approach risks are proposed and integrated using KF. The risk-state MPC is presented based on the landing risk gradient. In [7], a framework is proposed for landing in an uncertain environment based on point cloud in a coarse to fine manner. It has four modules: preprocessing point cloud, selection of course landing site, evaluation of fine terrain, and optimal landing model. An ACL problem is solved in [8] with input constraints and external disturbance. A relative motion model between the ideal glide path and the UAV is established, and the wave disturbance to the carrier is considered an external disturbance. A backstepping control law is proposed having input constraints, and the Lyapunov function is used to prove the stability of the closed-loop system.

In [9], deep-stall landing for the fixed wing UAV is proposed. The UAV is guided along the predefined path and performs landing at low speed with good precision. The landing control of high-speed UAV under wind interference is presented in [10]. Monte Carlo simulations show the system's robustness under wind disturbance. In [11], an automatic landing technique is presented which uses an $\mathcal{H}_2/\mathcal{H}_\infty$ technique along with a dynamic inversion method. An optimal observer is considered to tackle sensor errors and other disturbances. The design aims to develop a landing control system that not only cancels the negative effects of sensor errors and disturbances but also gives a good result when the number of states is more than the number of sensors. PID controllers (conventional and fuzzy variant) are compared with the dynamic inversion concept in [12]. There are three PID controllers for pitch, altitude, and velocity. Controllers are tested for gyro sensor error and wind shear. Numerical simulations validate the results. In [1], a Sliding Mode Control (SMC) technique is designed for autonomous landing. The landing maneuver is divided into glide path capture and flare phase. A straight line and an exponential curve specify both phases in longitudinal landing plane. The obtained controller is validated by simulating the landing maneuver using a nonlinear model with a large offset in the initial position from the reference landing trajectory. The results are compared with a conventional PID controller.

A hierarchical control structure for the autonomous landing is developed in [13]. Active disturbance rejection for attitude control, proportional guidance law for height tracking, and PID is used for heading angle, flight path angle, and taxi control. Hardware-In-the-Loop (HIL) simulation and field experiments are conducted to demonstrate the performance of the entire test system and the proposed approach. An intelligent landing system based on an Artificial Neural Network (ANN) is given in [2]. The weights are searched using the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) algorithm. The ANN is used to control the pitching torque and thrust of the UAV to obtain a smooth landing. The same algorithm is used in the case of high wind conditions. All possible flight conditions must be considered in the training data for efficient training of ANN. A fuzzy logic-based autonomous landing technique of UAV is proposed in [14]. Three fuzzy logic modules control the UAV's speed, altitude global position (longitude-latitude) against the runway. An SMC with a Cerebellar Model Articulation Controller (CMAC) for the landing of a UAV is presented in [15]. The SMC's parameters are adjusted using a Genetic Algorithm, Chaotic Particle Swarm Optimization (CPSO), and Particle Swarm Optimization (PSO). An intelligent landing scheme based on different CAMCs is presented in [16]. The Lyapunov theory is used for stability analysis and adaptive learning rules.

These techniques provide closed-loop performance, stability, and a certain degree of robustness but generally do not handle the constraints. Some modifications, e.g., anti-windup schemes, can be introduced to handle the constraints on input saturation. However, these make the design complicated (especially for multi-variable systems), limited to a restricted class of constraints, and may yield reduced closed-loop performance [17]. A more systematic way to handle the constraints is the Model Predictive Control (MPC) strategy. At each time step, the MPC uses the current state's information and predicts the system's evolution over a desired future horizon. Accordingly, the MPC designs the best input control sequences resulting in constraint satisfaction and the best performance. However, as mentioned in [18], it is more difficult to characterize the MPC's frequency-domain properties, robustness, and stability compared to many other linear state feedback techniques; this reduces the transfer of MPC technique to applications [19].

In the literature, [17,20,21], different techniques are presented for the selection of the objective/cost function matrices (P, Q and R) of the linear MPC controller in such a way that it behaves like any well-designed linear controller (e.g., H_∞ controller for our case) when the constraints are inactive. Hence, for any disturbance around the equilibrium point for which the inputs and states remain within the admissible range, the closed-loop properties of the MPC match the H_∞ synthesis. The advantage of H_∞-based MPC is that, contrary to H_∞, the resulting technique can explicitly handle the constraints during the transients along with the frequency domain properties of the H_∞ controller when the constraints are inactive. To obtain the cost function matrices, which are based on H_∞ controller, an LQR-based inverse optimization problem is formulated and solved in this work. Now, one can design the cost function matrices (P, Q and R) more sensibly.

Once the objective function has been designed, state information is required at each time step. Like many other applications, the full state information for landing control is unavailable. A state observer is required to initialize the prediction model. It is well-known that the Kalman Filter (KF) can be used to estimate the state in a full state feedback system, e.g., in Linear Quadratic Gaussian (LQG), which is the basis of MPC due to quadratic objective function. The normalized coprime factor robust stabilization approach provides a controller that can be written as plant observer form and have an optimized stability margin [22].

Most UAVs accidents that occur during autonomous landing are due to severe weather conditions. There are many natural uncertainties and disturbances such as wind gusts, wind shear, sensor noise, parametric uncertainties, etc that a UAV must deal with during a flight. In the worst-case scenario, the UAV might diverge from the reference path, e.g., in [23], where the path error reaches about 39 ft (approx. 12 m) and can result in a hard landing or runway overrun.

The main objective of this work is to design a robust control strategy for the autonomous landing phase of UAVs under wind disturbances. We propose the H_∞-based model predictive control for the UAV during the landing phase. The controller's performance is evaluated against moderate and severe wind disturbance, and results are compared with the benchmark study presented in [23].

The paper is organized as follows. In Section 2, the mathematical models for landing trajectory, UAV and windshear are presented. In Section 4, the design steps for H_∞-based MPC are explained. The MPC is realized for the landing control of UAV in Section 4. The results are discussed in Section 5 and conclusions are drawn in Section 6.

2. Mathematical Models of the System

In addition to a model of the UAV dynamics, a mathematical model of the landing trajectory, as shown in Figure 1, is also necessary. It is the reference path that the UAV will follow during landing. The performance of the UAV is affected by wind disturbances, and a wind model is needed to calculate this effect. This section presents the mathematical model of the landing trajectory, the non-linear and linear UAV, and the wind-shear model.

2.1. Landing Trajectory Model

There are two phases in a typical landing: glide and flare. During the glide phase, the UAV descends with the glide path angle of −3° to −4°. When the UAV reaches an altitude of about 30 m, the flare maneuverability is executed. The descent rate of the UAV is decreased for a comfortable and smooth touchdown. For the glide phase ($h \geq h_0$, h_0 is the altitude at which the glide phase ends and flare begins), the reference altitude h_r and actual altitude h are:

$$\begin{aligned} \dot{h}_r &= V_0 \sin\gamma_r \approx V_0 \gamma_r \\ \dot{h} &= V_0 \sin\gamma \approx V_0 \gamma \end{aligned} \tag{1}$$

where $\gamma = \theta - \alpha$ is the actual glide path angle of the UAV during the first phase of landing and $\gamma_r = \theta_r - \alpha$ is the reference glide path angle, θ is the pitch angle, α is the angle of attack and V_0 is the nominal flight speed of the UAV. Generally $\gamma_r \approx 3^0$ hence the small angle approximations are justified in Equation (1). The dynamics of the flare phase are presented in detail in [11]. The altitude h_r is an exponential decay function represented as $h_r = h_0 \exp -t/\tau$, with τ is the time constant that define the exponential curve of flare. The derivative of the above equation is:

$$\dot{h}_r = -\frac{1}{\tau} h_0 \exp -t/\tau = -\frac{1}{\tau} h_r. \tag{2}$$

By using Equations (1) and (2), the reference pitch angle θ_r for both phases can be determined:

$$\theta_r \;\approx\; \alpha + \frac{\dot{h}_r}{V_0} \approx \begin{cases} \alpha + \gamma_r, & h \geq h_0, \\ \alpha - \dfrac{1}{V_0 \tau} h_r, & h < h_0. \end{cases} \tag{3}$$

2.2. UAV Model

The UAV is assumed to be a rigid body in still air with decoupled lateral and longitudinal dynamics. Since the landing is mostly a longitudinal process, only the longitudinal dynamics are considered in this study. The longitudinal dynamics are given as follows [24]:

$$\begin{aligned} \dot{U} &= -\,QW - g_0 \sin\theta + \frac{F_x}{m} \\ \dot{W} &= QU + g_0 \cos\theta + \frac{F_z}{m} \\ \dot{\theta} &= Q \\ \dot{Q} &= \frac{M_y}{J_y} \end{aligned} \tag{4}$$

where U is the longitudinal body-axes velocity, W is the vertical body-axes velocity, θ is the pitch, Q the is pitch rate, g_0 is the gravitational acceleration, m is mass, and J_y is the moment of inertia in the pitch of the UAV. F_x and F_z are the aerodynamic forces along the x- and z-axis, respectively and M_y is the pitching moment about the y-axis. These forces and moments are defined as:

$$\begin{aligned} F_x &= \frac{1}{2} \rho V_0^2 S C_x + T \\ F_z &= \frac{1}{2} \rho V_0^2 S C_z \\ M_y &= \frac{1}{2} \rho V_0^2 S \bar{c} C_m \end{aligned} \tag{5}$$

here $V_0 = \sqrt{U^2 + W^2}$, ρ is air mass density, T is thrust, S is wing area and $\bar{c}$ is mean aerodynamic chord. In Equation (5), the aerodynamic coefficients can be defined as:

$$\begin{aligned} C_x &= C_{x_0} + C_{x_\alpha} \alpha + C_{x_{\delta_e}} \delta_e \\ C_z &= C_{z_0} + C_{z_\alpha} \alpha + C_{z_{\delta_e}} \delta_e \\ C_m &= C_{m_0} + C_{m_\alpha} \alpha + C_{m_{\delta_e}} \delta_e + \frac{C_{m_q} Q \bar{c}}{Vel} \end{aligned} \tag{6}$$

In this work, a medium-sized UAV is used as a test vehicle. The physical parameters and aerodynamic coefficients of the test vehicle are given in Tables 1 and 2 respectively.

Table 1. Physical Parameters of the Test Vehicle.

Parameter	Value	Description
m	350 kg	Mass
J_y	300 kg·m^2	Moment of inertia about y-axis
S	6.5 m^2	Wing area
$\bar{c}$	6.6 m	Mean aerodynamic chord

Table 2. Aerodynamic Coefficients of the Test Vehicle.

Parameter	Value	Parameter	Value
C_{x_0}	−0.031	C_x^α	−0.088
$C_{x_{\delta_e}}$	−0.01	$C_z^{\delta_e}$	−0.124
C_{z_0}	−0.129	C_z^α	−3.368
C_{m_0}	0.003	C_m^α	−0.4
C_{m_q}	−2.0	$C_{m_{\delta_e}}$	0.25

The non-linear model given in Equation (4) is linearized by numerically perturbing the states and inputs about the operating point. The algorithm introduces a small perturbation to the states and inputs, one at a time, measures the response of the system and computes the state-spaces matrices. The longitudinal model has four states. However, the states can be expanded to include x-horizontal distance and h-altitude of the UAV as:

$$\begin{aligned} \dot{x} =& \ U\text{cos}\theta + W\text{sin}\theta, \\ \dot{h} =& \ U\text{sin}\theta - W\text{cos}\theta. \end{aligned} \tag{7}$$

The non-linear model is trimmed for the conditions of $U = 50$ m/s, $W = 3$ m/s, $\theta = 0.5$ deg, $Q = 0$ deg/s, $h = 300$ m and initial position $x = 0$ m. The inputs to the system is elevator deflection δ_e and thrust δ_{th}. The state-space model is written as follows:

$$\begin{aligned} \dot{x}_v &= A_v x_v + B_v u_v \\ y_v &= C_v x_v \end{aligned} \tag{8}$$

where $x_v = [u \; w \; \theta \; q \; h \; x]$ is the perturbed state vector, $u_v = [\delta_e \; \delta_{th}]$ is the perturbed input vector and $y_v = [u \; q \; \theta \; h]$ is the perturbed output vector. A_v, B_v and C_v are the state space matrices of the vehicle model and are given in Appendix A.

2.3. Wind Shear Model

Wind shear is a rapid change in wind direction or speed over a short distance or time. In the case of a macro-burst, the wind shear diameter is over 4 km, whereas a micro-burst is smaller than 4 km. A micro-burst may last a few seconds, but its effects, such as variation and extreme speed, can be very dangerous for UAVs.

Various models of micro-bursts have been developed. One is the vortex-ring model, proposed by Wood and Woodfield [25]. The vortex ring induces the velocity wind, and the micro-burst is represented by the two rings symmetrically to satisfy the boundary conditions. One is called the primary vortex ring above the ground, and the other is of the same strength below the ground and is called the imaginary vortex ring. Figure 2 illustrates the vortex-ring model.

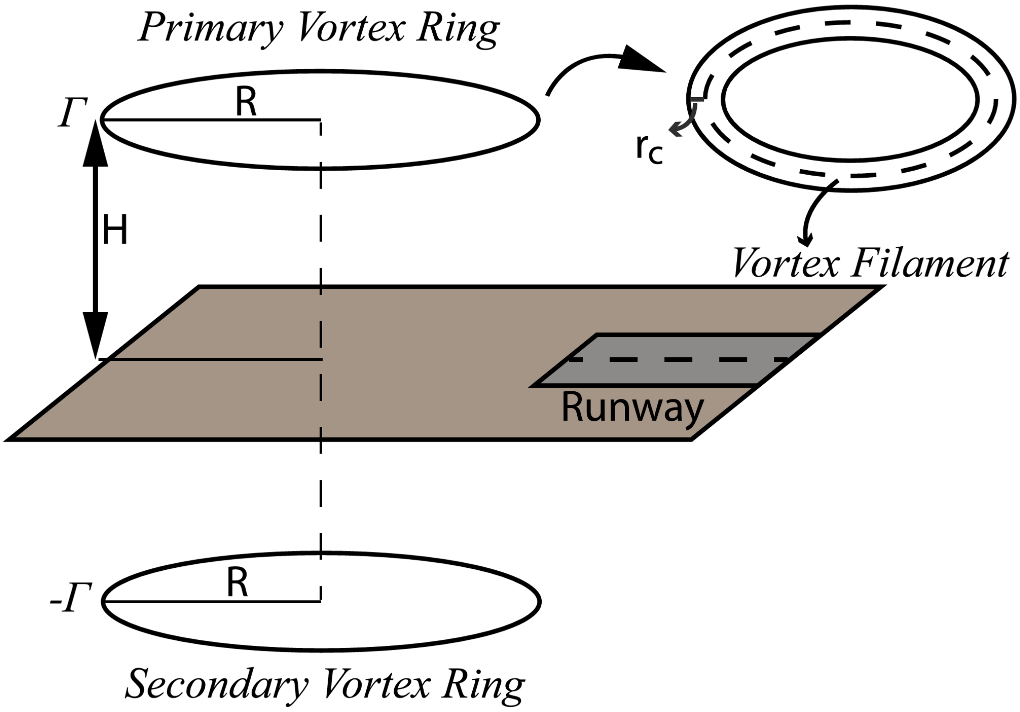


Figure 2. Vortex-Ring Model [23].

Parameters r_C, R, and Γ represent the radius of the finite core, and the radius of the vortex-ring and vortex-ring model circulation, respectively. The primary ring's centre coordinates are X, Y, and H. Multiple pairs of the ring can enhance the accuracy of the wind shear. A simplified vortex-ring down-burst model with two rings is presented in [23] and is used in this work. We can take $Y = 0$ because only longitudinal dynamics are considered in our study. If h and x are vertical and horizontal points of interest, respectively, then the induced velocities are computed as follows:

$$\begin{aligned} x_1 &= x - X - R & x_2 &= x - X + R \\ h_p &= h - H & h_m &= h + H \\ r_{1p} &= x_1^2 + h_p^2 & r_{2p} &= x_2^2 + h_p^2 \\ r_{1m} &= x_1^2 + h_m^2 & r_{2m} &= x_2^2 + h_m^2 \\ r_0 &= \min\{r_{1p}, r_{2p}\} & \zeta &= 1 - \exp^{-(r_0/r_c^2)} \\ r_{xp} &= \sqrt{(x - X)^2 + h_p^2 + R^2} \\ r_{xm} &= \sqrt{(x - X)^2 + h_m^2 + R^2} \\ r_{hp} &= \left[(x - X)^2 + h_p^2 + R^2\right]^{3/4} \\ r_{hm} &= \left[(x - X)^2 + h_m^2 + R^2\right]^{3/4} \end{aligned} \tag{9}$$

If $r_0 < \epsilon$, where ϵ is a small number that represents a point close to the ring filament, then,

$$W_x = 0, \qquad W_h = 0$$

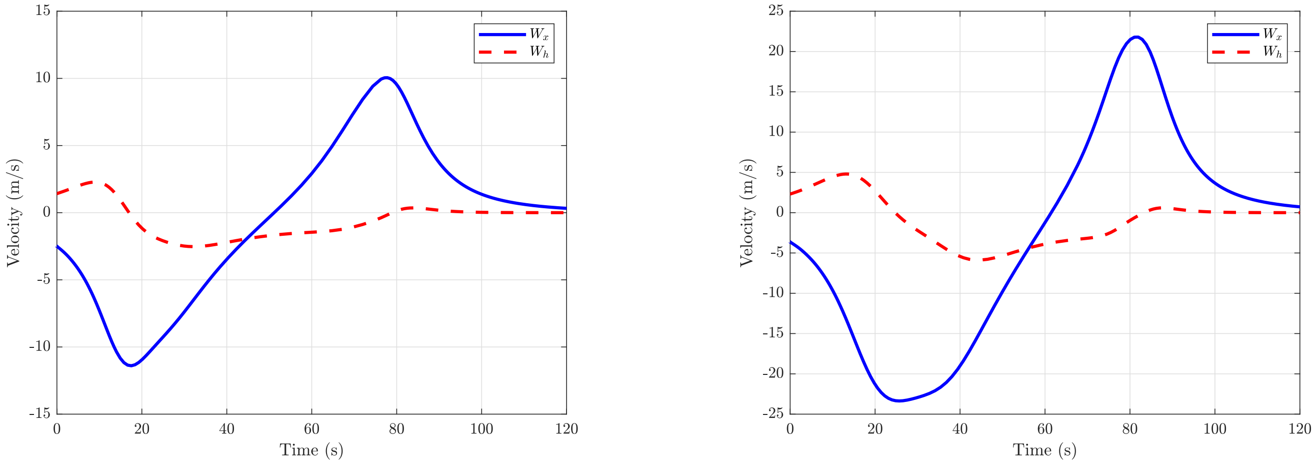
Otherwise,

$$\begin{aligned} W_x &= \frac{1.182\Gamma\zeta}{2\pi}\left[\frac{R}{r_{xp}}\left(\frac{h_p}{r_{2p}}-\frac{h_p}{r_{1p}}\right)-\frac{R}{r_{xm}}\left(\frac{h_m}{r_{2m}}-\frac{h_m}{r_{1m}}\right)\right] \\ W_h &= \frac{1.576\Gamma\zeta}{2\pi}\left[\frac{R}{r_{hp}}\left(\frac{x_1}{r_{1p}^{3/4}}-\frac{x_2}{r_{2p}^{3/4}}\right)-\frac{R}{r_{hm}}\left(\frac{x_1}{r_{1m}^{3/4}}-\frac{x_2}{r_{2m}^{3/4}}\right)\right] \end{aligned} \quad (10)$$

where W_h and W_x are vertical and horizontal induced velocities, respectively. Two sets of parameters are shown in Table 3, which are used to calculate W_h and W_x for simulations. The first set of parameters represents a moderate down-burst and the second for severe down-burst. It is assumed that the UAV encounters the down-burst at the altitude of 300 m. Moderate and severe down-bursts are shown in Figure 3.

Table 3. Downburst Parameters for Two Rings

Parameter	Moderate	Severe	Unit
Γ_1	18,580	37,160	m^2/s
R_1	1676	1524	m
H_1	610	610	m
R_{c1}	152	152	m
Γ_2	11,148	26,013	m^2/s
R_2	1220	1067	m
H_2	762	610	m
R_{c2}	152	91	m



(**a**) Moderate Downburst

(**b**) Severe Downburst

Figure 3. Windshear Effects.

Simulation models of UAVs need to incorporate the above mathematical representation of wind-shear effects. After embedding the wind-shear model in the Equations (4) and (7), the UAVs model is as follows [23]:

$$\begin{aligned} \dot{U} &= -QW - g_0\sin\theta + \frac{F_x}{m} + \dot{W}_x \\ \dot{W} &= QU + g_0\cos\theta + \frac{F_z}{m} + \dot{W}_h \\ \dot{x} &= U\cos\theta + W\sin\theta + W_x \\ \dot{h} &= U\sin\theta - W\cos\theta + W_h \end{aligned} \quad (11)$$

3. H_∞ Based MPC Framework

The H_∞-based MPC methodology has the following design steps [26].

1. Design the input and output weights (W_1 and W_2) for the model to meet the desired closed-loop specifications. It is the basic step of the H_∞ loop shape design procedure as explained in [27].
2. Synthesize the H_∞ compensator. It has an observer-based state feedback structure. As a result, we have controller matrix K and observer matrix H, which are used to calculate the state matrix to initialize the prediction model at each time step.
3. By using the controller matrix K, from Step 2, formulate and solve an inverse optimization problem to design cost function matrices $P,\ Q$, and R.
4. Solve the MPC problem at each time step using the cost function matrices designed in step 3.

This methodology inherits the transient response, reference tracking, disturbance rejection, and stability margin of the H_∞ controller in the region where the constraints are inactive. As the observer is designed by H_∞ synthesis, it does not require additional tuning. The H_∞ loop shaping control methodology of [28] in which the controller can be realized as an observer plus a state feedback control law is given in the next subsection.

3.1. H$_\infty$ Loop Shaping Design using Observer Structure

In general, H_∞ controllers do not have an explicit structure [29]. However, the normalized coprime factor robust stabilization method provides a controller that can be written in the form of a plant observer [30]. The resulting controller can be written as an exact plant observer and state feedback:

$$\begin{aligned} \dot{\hat{x}}_s &= A_s\hat{x}_s + H_s(C_s\hat{x}_s - y_s) + B_s u_s \\ u_s &= K_s\hat{x}_s, \end{aligned} \tag{12}$$

where $A_s,\ B_s$, and C_s are the state-space realizations of the shaped plant, $\hat{x}_s$ is the observer state vector, u_s is the input, and y_s is the output vector of the shaped plant, and

$$\begin{aligned} H_s &= -Z_sC_s^T \\ K_s &= -B_s^T\left[I - \gamma^{-2}I - \gamma^{-2}X_sZ_s\right]^{-1}X_s \end{aligned} \tag{13}$$

where Z_s and X_s are the solutions of the following algebraic Riccati equations,

$$\begin{aligned} A_sZ_s + Z_sA_s^T - Z_sC_s^TC_sZ_s + B_sB_s^T = 0 \\ A_s^TX_s + X_sA_s - X_sB_sB_s^TX_s + C_s^TC_s = 0\,. \end{aligned} \tag{14}$$

The maximum stability margin is calculated with $\epsilon_{\max} = \frac{1}{\gamma_{\min}}$, where $\gamma_{\min} = \sqrt{1+\rho(X_sZ_s)}$. where ρ is the maximum eigenvalue of matrix X_sZ_s. If $\gamma_{\min} < 4$ the design is usually considered successful. The design steps for obtaining K_s and H_s are explained in the next subsections.

3.2. Design of H$_\infty$ Controller

The linear model $G_v : C \mapsto C^{2\times 4}$ is used to describe the design methodology. To meet the desired specification following pre- and post-compensators are selected,

$$W_1 = \mathrm{diag}\left(\frac{3(s+1)}{s},\ \frac{(s+1)}{s}\right) \tag{15}$$

$$W_2 = \mathrm{diag}\left(1,\ 1.5,\ \frac{(s+0.01)}{s},\ \frac{1.2(s+0.01)}{s}\right) \tag{16}$$

The resulting shaped plant is obtained as:

$$G_{\text{sp}} = W_1\, G_{\text{v}}\, W_2 \tag{17}$$

After including the weights, the state dimensions of the shaped plant have been increased. The state-space matrices A_s, B_s and C_s of the shaped plant are given in Appendix B. Following the procedure of H_∞ loop shaping control, as explained in Section 3.1, the controller K_s and observer H_s are synthesized and given in Appendix B. The normalized coprime factor robust optimization gives $\gamma = 2.81 < 4$, which meets the desired criteria. The singular values of the plant (vehicle model), shaped plant and shaped plant with controller are shown in Figure 4.

The designed controller K_s is used to formulate and solve the inverse optimization problem which is explained in the next subsection.

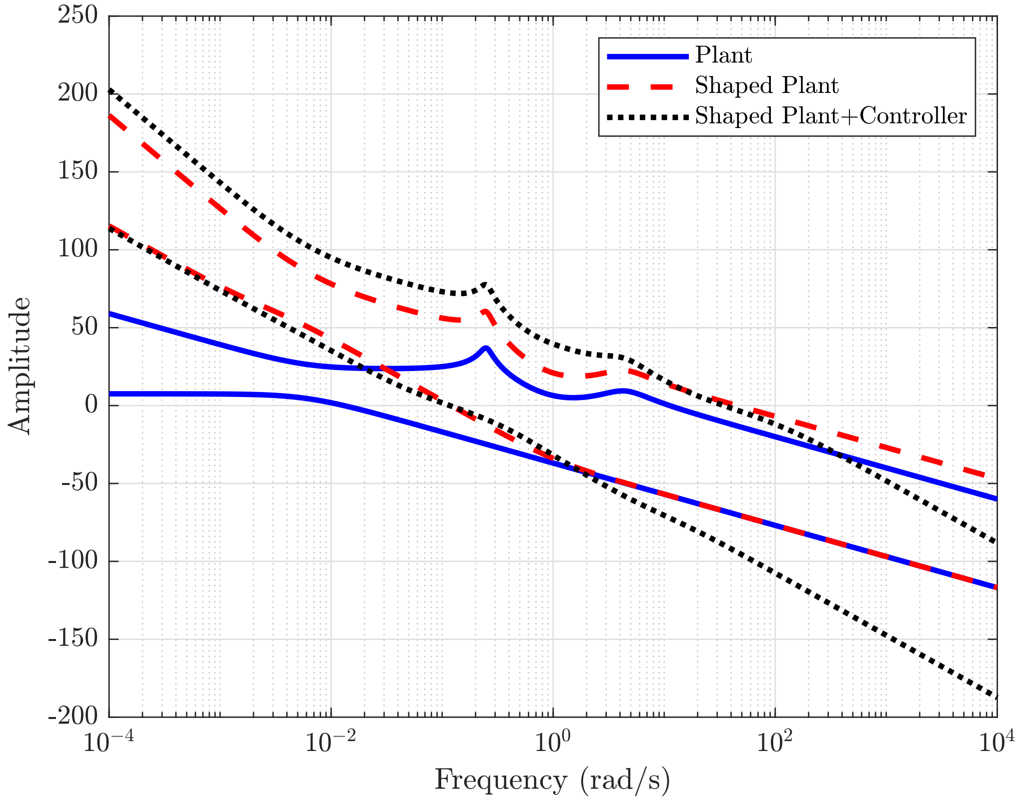


Figure 4. Singular Values.

3.3. Inverse Optimal Problem

The optimal control theory uses the linear quadratic regulator (LQR) to find the infinite horizon full-state feedback control law for both continuous and discrete LTI systems. The formulation for the shaped discrete-time LTI system is,

$$x_{k+1} = A_s x_k + B_s u_k \tag{18}$$

which minimizes the objective function

$$J = \sum_{k=0}^{\infty} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} Q_s & 0 \\ 0 & R_s \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}. \tag{19}$$

The optimal feedback control is

$$u_k = -K_{\mathrm{LQR}} x_k, \tag{20}$$

where,

$$K_{\mathrm{LQR}} = (B_s^T P_s B_s + R_s)^{-1} B_s^T P_s A_s \tag{21}$$

Here, P_s is the unique positive semi-definite solution of the discrete-time algebraic Riccati equation (DARE) given as:

$$A_s^T P_s A_s - P_s - (A_s^T P_s B_s)(B_s^T P_s B_s + R_s)^{-1} B_s^T P_s A_s + Q_s = 0 \tag{22}$$

where, Q_s is semi-positive definite and R_s is a positive-definite matrix. Now, if this problem is solved in the reverse order by assuming that we have an optimal feedback controller K_{LQR} and our objective is to find the cost function matrices for the shaped plant. The methods for solving this problem are explained in the coming subsection.

Solution of Inverse Optimal Problem

The basic aim is to find the cost function matrices (P_s, Q_s, and R_s). The inverse optimal problem is formulated with Equations (21) and (22) by taking $K_{\mathrm{LQR}} = K_s$. The resulting equations become as follows:

$$\begin{aligned} A_s^T P_s A_s - P_s - (A_s^T P_s B_s) K_s + Q_s &= 0 \\ B_s^T P_s A_s - (B_s^T P_s B_s + R_s) K_s &= 0 \end{aligned} \tag{23}$$

In Equation (23), there are two equations and three unknowns (P_s, Q_s and R_s). Reference [26] gives an analytical solution to the above problem by taking $R_s = I$. It may yield numerically ill-conditioned cost function matrices, which result in an ill-conditioned real-time optimization problem. An LMIs-based solution to this problem is presented in [31]. An additional criterion is defined such that the optimal solution must reduce the condition number of the cost function matrices. The LMIs-based optimization problem for the discrete-time system is defined as follows:

$$\begin{aligned} (\hat{Q}_s,\ \hat{R}_s,\ \hat{P}_s,\ \hat{\alpha}) = \arg \min_{Q_s, R_s, P_s, \alpha} \alpha^2&, \text{ such that} \\ P_s &\geq 0 \\ A_s^T P_s A_s - P_s - (A_s^T P_s B_s) K_s + Q_s &= 0 \\ B_s^T P_s A_s - (B_s^T P_s B_s + R_s) K_s &= 0 \\ I \leq \begin{bmatrix} Q_s & 0 \\ 0 & R_s \end{bmatrix} &\leq \alpha I \end{aligned} \tag{24}$$

The LMI optimization problem (24) can be solve efficiently using the SeDuMi [32] package with the YALMIP modelling toolbox [33] in MATLAB 2016. The optimal estimated cost function matrices are given in Appendix B. In contrast to the standard procedure for selecting cost function matrices, where usually diagonal terms are used based on performance, here we have an off-diagonal term. Now, these matrices are used for the realization of the MPC.

4. MPC Realization

MPC is a feedback control methodology in which an optimization problem is solved online. At each time step, the optimization algorithm computes the control input sequence over a finite-time horizon that minimizes the objective function subject to constraints on input, output, and/or state. The first element of the sequence is applied to the system on that sample time, and the process is repeated on every sampling step in a recursive, receding horizon manner. (The subscript "s" is ignored in the rest of the work since we are designing MPC for a shaped system.)

The MPC control methodology is based on the following steps.

1. A prediction model of the UAV,

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \tag{25}$$

where x_k is the state of the plant, u_k is the control input of the system and y_k is the output of the system at time kT, where T is the sample time.

2. An objective function,

$$J(u) = x_N^T P x_N + \sum_{k=1}^{N-1} x_k^T Q x_k + u_k^T R u_k \tag{26}$$

The matrices $P > 0,\ Q > 0$, and $R > 0$ are designed to meet the desired performance requirements, and N is the prediction horizon.

3. The inequality constraints,

$$v_{\min} \leq v_k \leq v_{\max}, \tag{27}$$

here, v_k are the constraint inputs, outputs and/or states and $v_{\min}$ and $v_{\max}$ are maximum and minimum limits on constraints.

4. An optimization algorithm to minimize the objective function.

The MPC is realized for the shaped model having input constraints $[-25°,\ 25°]$, and $[0,\ 100\%]$ for elevator deflection and percentage thrust, respectively. The dense approach presented in [34] is used for the given LTI systems. The cost function and prediction model with prediction horizon N can be written as follows:

$$J(x, x_0) = x_0^T Q x_0 + \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}^T \underbrace{\begin{bmatrix} Q & 0 & 0 & \cdots & 0 \\ 0 & Q & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & Q & 0 \\ 0 & 0 & \cdots & 0 & P \end{bmatrix}}_{\bar{Q}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}$$

$$+ \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}^T \underbrace{\begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R \end{bmatrix}}_{\bar{R}} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\bar{T}} x_0 + \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}}_{\bar{S}} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$$
\begin{aligned}
J(z, x_0) &= x_0^T Q x_0 + (\bar{T}x_0 + \bar{S}z)^T \bar{Q}(\bar{T}x_0 + \bar{S}z) + z^T \bar{R} z \\
&= \frac{1}{2} z^T \underbrace{2(\bar{R} + \bar{S}^T \bar{Q}\bar{S})}_{H} z + x_0^T \underbrace{2\bar{T}^T \bar{Q}\bar{S}}_{F} z \\
&= \frac{1}{2} z^T H z + x_0^T F z
\end{aligned}
$$

where z is the input vector. As we are designing the control system for the shaped plant, it is necessary to convert the input constraint limits from u_v to u_s. The constraints matrices for the shaped plant are as follows:

$$
\underbrace{\begin{bmatrix} C_{w1}B_{w1} & 0 & \cdots & 0 \\ C_{w1}A_{w1}B_{w1} & C_{w1}B_{w1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_{w1}A_{w1}^{N-1}B_{w1} & C_{w1}A_{w1}^{N-2}B_{w1} & \cdots & B_{w1} \end{bmatrix}}_{G} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ x_{N-1} \end{bmatrix} \leq \begin{bmatrix} u_{\max} \\ u_{\max} \\ \vdots \\ u_{\max} \end{bmatrix} - \underbrace{\begin{bmatrix} C_{w1}A_{w1} \\ C_{w1}A_{w1}^2 \\ \vdots \\ C_{w1}A_{w1}^N \end{bmatrix}}_{S} x_{w1}
$$

The interior point method minimizes the above cost function along with constraints at each time step. The detailed results are presented and discussed in the next section.

5. Results and Discussion

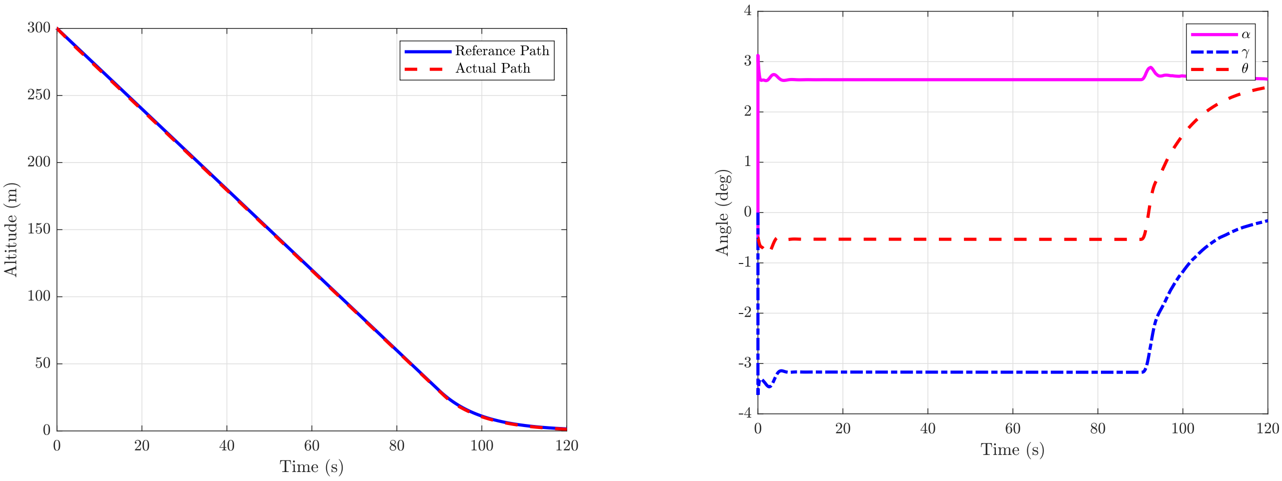
The proposed control law is implemented in a non-linear simulation environment to validate the designed algorithm. The simulation was conducted for two different scenarios. In the first case, the design is tested for the reference tracking in an ideal environment and the second case illustrates the disturbance rejection property of the design. The windshear effects given in Section 2.3 are used as external disturbances, and results are compared to the study presented in [23].

5.1. Case 1

In the first case, the linear model of the UAV given in Equation (8) is used to ensure that it follows the reference trajectory under ideal conditions. The altitude error is minimized by elevator deflection and thrust demand. Figure 5a shows that the actual path closely follows the reference path. The result also indicates that the angle of attack, pitch, and flight path angle have be smoother, as shown in Figure 5b. The Figure 5c,d shows the elevator and thrust demand respectively.

5.2. Case 2

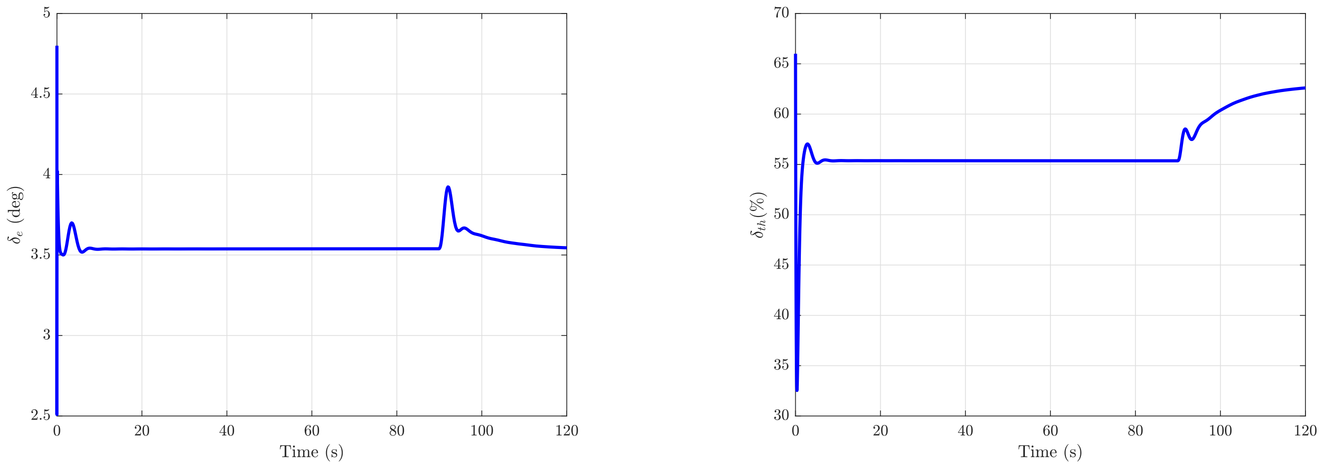
In second case, we assessed the design algorithm under moderate and severe winds-hear. The UAV encounters the wind at the start of the glide slope and its effects last until the flare ends. It is assumed that initial altitude of the UAV is 300 m and the flare phase begins at an altitude of 30 m. In Equation (11), winds-shear components were introduced into the dynamic equations of the UAV. The equation shows that altitude depends on U, W, θ, and W_h. The wind shear component W_h is a disturbance input that cannot be controlled, however the controller can compensate for the wind shear effects using U, W, and θ. The controller dynamically adjusts θ and hence γ to keep the UAV on the desired path. Contrary to the ideal case, it is not required to maintain a constant angle of attack α in the presence of the wind shear.



(**a**) Altitude Level

(**b**) Angle of Attack (α), Glide Path Angle (γ) and Pitch Angle (θ)

Figure 5. *Cont.*

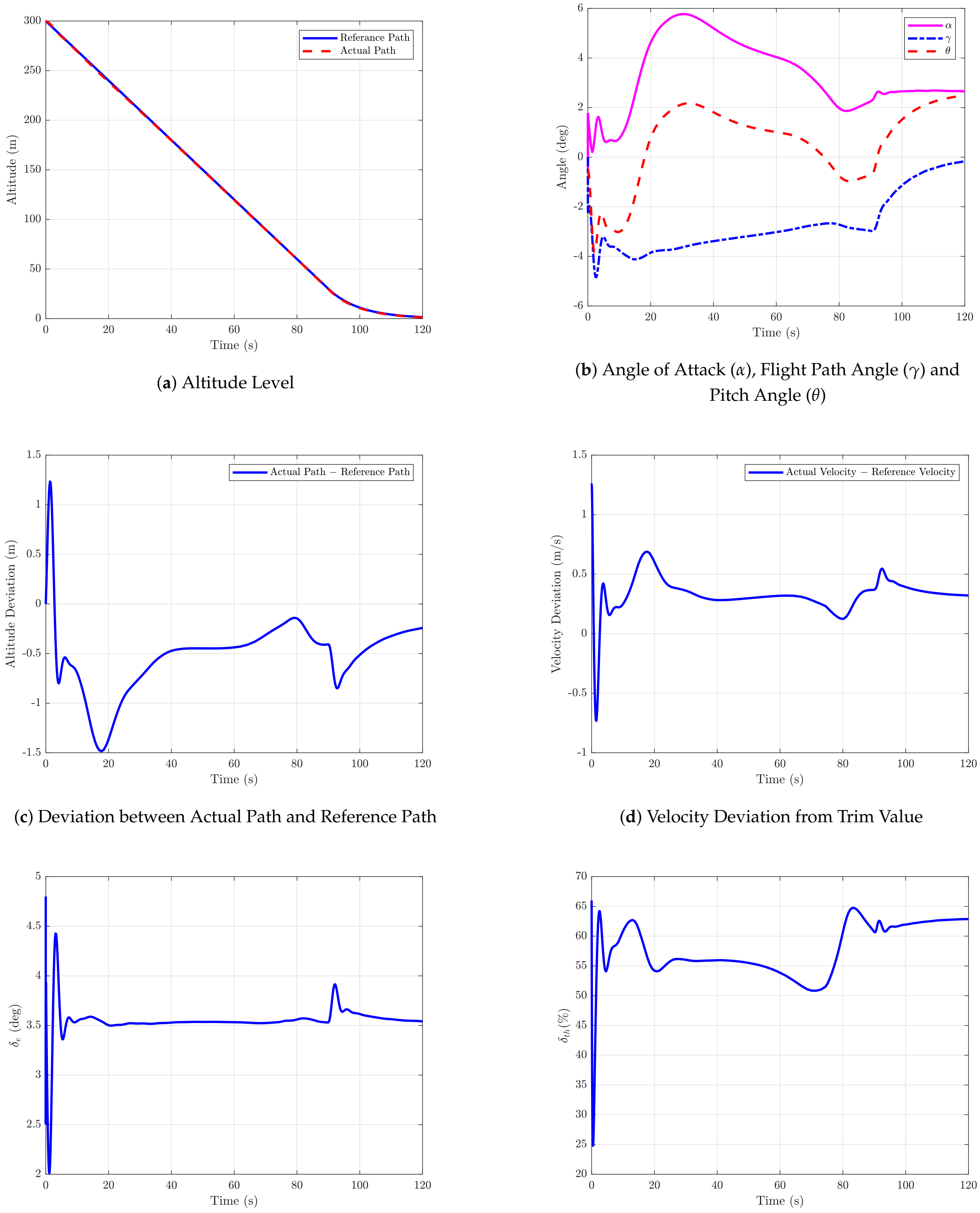


(**c**) Elevator Deflection

(**d**) Thrust Demand

Figure 5. Case 1: No Wind Effect.

Figure 6a shows that the headwind increase during the first 20 s and the UAV has increased the longitudinal velocity component to compensate its effects. Figure 6b shows that the angle of attack also increases in response to the vertical wind component. As a result, the altitude and velocity return to the desired level with little deviation as shown in Figure 6c,d respectively. After 20 s, the headwind effects start to weaken and the angle of attack decreases to maintain the desired altitude. The UAV starts encountering a tailwind after 50 s which increases the longitudinal velocity and decreases the angle of attack. The controller reduces the velocity and increases the angle of attack by manipulating the thrust demand and elevator deflection. The flare phase begins after 90 s when the tailwind gradually loses its effects. Figure 5a show that the UAV follows the reference trajectory closely during both glide and flare phases and performs a smooth touchdown. The displacements of the elevator and throttle actuators remain within limits and are shown in Figure 6e,f. The controller is also tested under a severe downburst. The results in Figure 7 are similar to the moderate downburst case, but their magnitudes are greater. The UAV closely follows the reference trajectory and the elevator deflection angle and throttle position remain within limits.



(**a**) Altitude Level

(**b**) Angle of Attack (α), Flight Path Angle (γ) and Pitch Angle (θ)

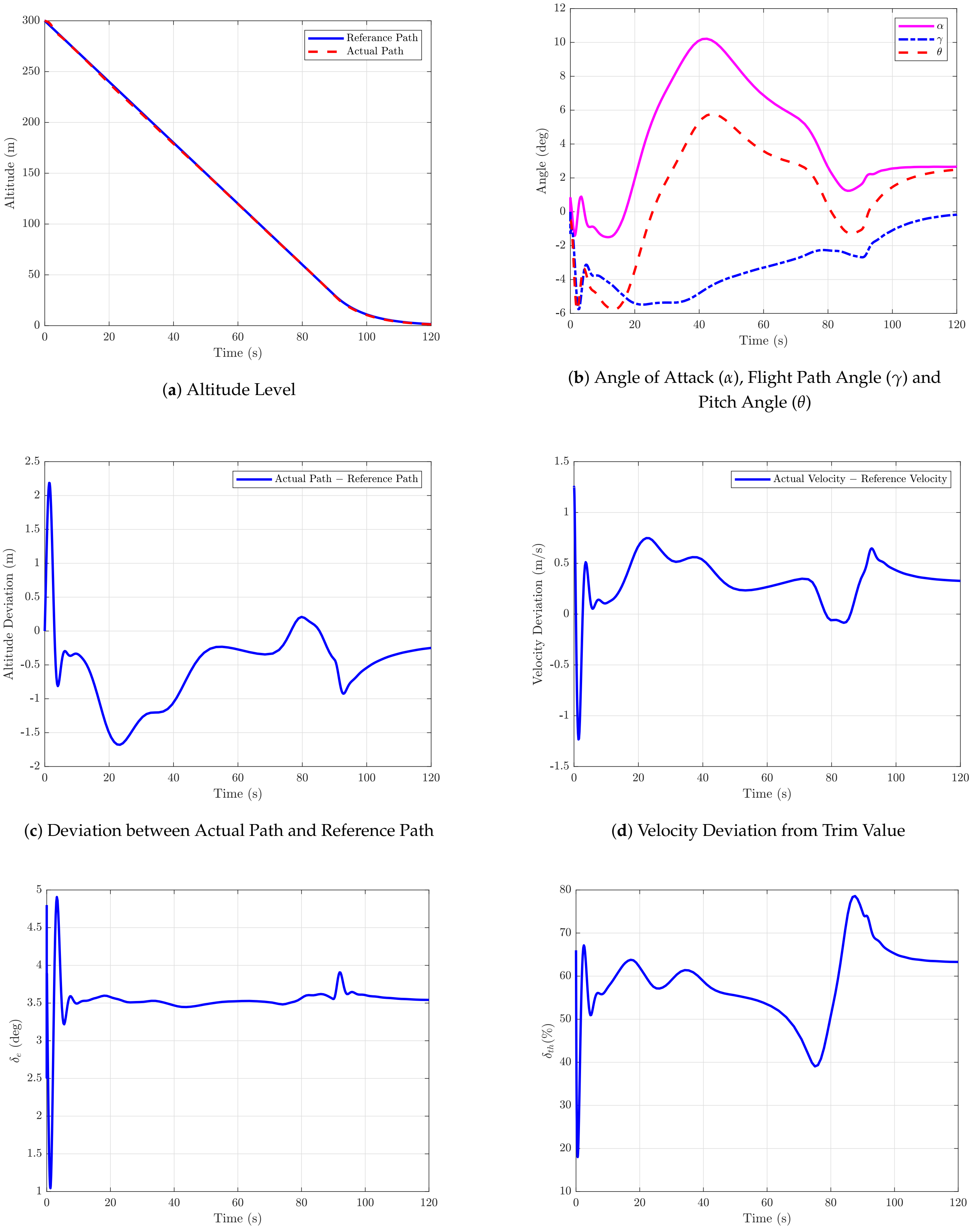
(**c**) Deviation between Actual Path and Reference Path

(**d**) Velocity Deviation from Trim Value

(**e**) Elevator Deflection

(**f**) Thrust Setting

Figure 6. Case 2: Moderate Downburst.



(**a**) Altitude Level

(**b**) Angle of Attack (α), Flight Path Angle (γ) and Pitch Angle (θ)

(**c**) Deviation between Actual Path and Reference Path

(**d**) Velocity Deviation from Trim Value

(**e**) Elevator Deflection

(**f**) Thrust Setting

Figure 7. Case 2: Severe Downburst.

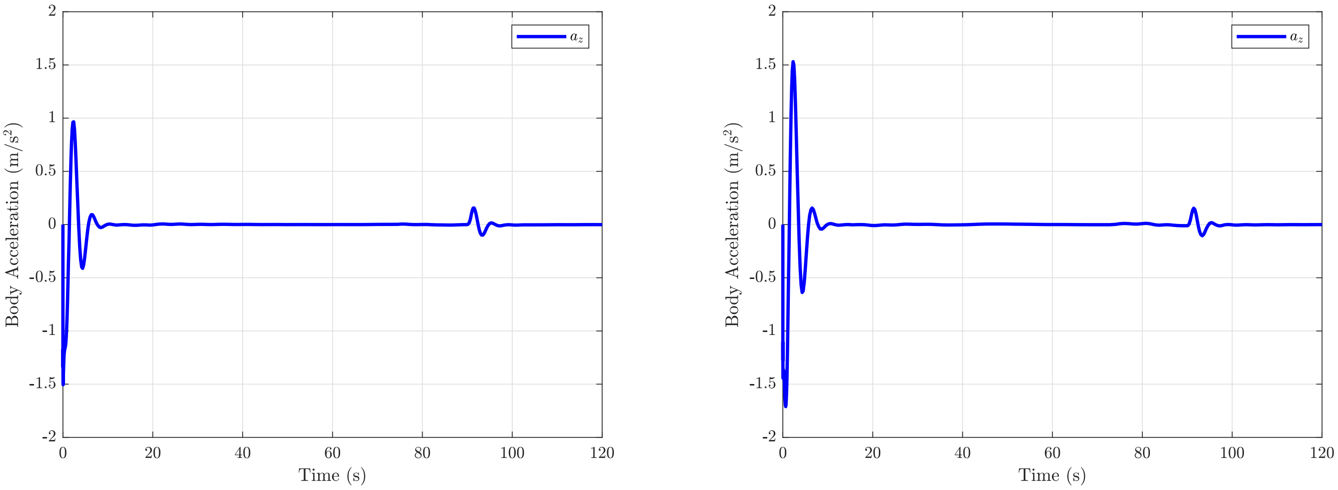
The proposed control methodology is compared with the study reported in [23]. Note that our UAV is different from the system in [23]. For a fair comparison, the wind load on both vehicles is calculated, and the controller performance is assessed under the severe downburst effects. The wind load F_{wind}, on the air vehicle is calculated by:

$$F_{\text{wind}} = \frac{1}{2}\rho V_{\text{wind}} S C_L \tag{28}$$

where ρ represents the air density, V_{wind} is the wind velocity, S is the wing reference area, and C_L is the lift coefficient. The acceleration a_{zw}, due to wind load is defined as:

$$a_{zw} = \frac{F_{\text{wind}}}{m} \tag{29}$$

As shown in the Table 4, the vehicle acceleration (a_{zw}) due to wind load is nearly the same for both systems. The controller did not react properly in the reference study, which worsened the situation. Contrary to this, our proposed controller reacts efficiently and cancels the wind effect significantly. Figure 8 shows the vehicle acceleration a_z throughout the landing trajectory.



(**a**) For Moderate Downburst (**b**) For Severe Downburst

Figure 8. Body Acceleration along *z*-axis.

Table 4. Comparison of Test Vehicle with Reference.

Parameter	Reference Vehicle [23]	Test Vehicle	Unit
Mass	73,482	350	kg
Wing reference area	201.6	6.5	m^2
Wind velocity	23	23	m/s
Wind load	63,988	268.19	N
a_{zw}(wind)	0.871	0.766	m/s^2
a_z(controller)	2.1	0.00627	m/s^2

6. Conclusions

A different approach for the landing fixed-wing UAVs is presented in this paper. Glide and flare reference trajectories are predefined and the UAV is forced to follow the altitude of the reference trajectory. First, the model is robustly stabilized using the normalized coprime factorization method and the controller K_s and observer H_s are obtained. Then, the LQR-based inverse optimal problem is formulated and solved to design the cost function matrices, which are further utilized to realize MPC. The designed MPC inherits the small-signal properties (stability margin and closed-loop performance) of the H_∞ controller, when the constraints are inactive (i.e., perturbation around equilibrium points that keep the system within saturation limits). Various scenarios are simulated and studied. The results demonstrate the proposed techniques' effectiveness and correctness even under moderate and severe wind shear effects. A qualitative analysis is also performed to compare the results with the benchmark work. The landing results demonstrated a significant improvement.

Author Contributions: Conceptualization, Z.L., A.I.B. and R.S.; methodology, Z.L., A.S. and A.I.B.; investigation, Z.L., A.S. and R.S.; writing—original draft preparation, Z.L, A.I.B. and J.F.W.; writing—review and editing, Z.L. and J.F.W.; visualization, Z.L.; supervision, A.I.B. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. State Space Matrices of Test Vehicle

The state space matrices from Equation (8) for the test vehicle A_v, B_v and C_v are given below:

$$A_v = \begin{bmatrix} -0.0399 & -0.0541 & -0.1710 & -0.0524 & 0 & 0 \\ -0.2783 & -1.9805 & -0.0045 & 0.8727 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1.1360 & -18.3577 & 0 & -1.9144 & 0 & 0 \\ 0.0087 & -1 & 0.8737 & 0 & 0 & 0 \\ 1 & 0.0087 & 0.0295 & 0 & 0 & 0 \end{bmatrix}$$

$$B_v = \begin{bmatrix} -0.0050 & -0.0598 & 0 & 9.9890 & 0 & 0 \\ 0.0143 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$C_v = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Appendix B. State Space, Controller and Observer Matrices of Shaped Plant

The state space matrices (A_s, B_s and C_s), controller gain (K_s), observer gain (H_s) and MPC cost function matrices (P_s, Q_s and R_s) for the shaped plant as as follows:

$$A_s = \begin{bmatrix} 0.9992 & -0.0008 & -0.0034 & -0.0011 & 0 & -0.0010 & 0.0014 \\ -0.0052 & 0.9586 & -0.0000 & 0.0168 & 0 & 0.0026 & 0 \\ 0.0002 & -0.0036 & 1.0000 & 0.0196 & 0 & 0.0099 & 0 \\ 0.0224 & -0.3508 & -0.0000 & 0.9594 & 0 & 0.9800 & 0 \\ 0.0002 & -0.0196 & 0.0175 & 0 & 1 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_s = \begin{bmatrix} -0.0041 & 0.0105 & 0.0397 & 3.9596 & 0.0002 & 0.0800 & 0 \\ 0.0058 & 0 & 0 & 0.0001 & 0 & 0 & 0.0800 \end{bmatrix}^T$$

$$C_s = \begin{bmatrix} 20 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 \end{bmatrix}$$

$$K_s = \begin{bmatrix} 0.0244 & -0.56 & 1.0539 & 0.2459 & 1.1537 & 0.2479 & 0.0001 \\ 19.1416 & -12.125 & 20.0018 & 0.0215 & 32.2626 & 0.0026 & 0.2311 \end{bmatrix}$$

$$H_s = \begin{bmatrix} -2.54\times10^{-9} & -2.37\times10^{-8} & -3.68\times10^{-8} & -1.21\times10^{-6} & -2.54\times10^{-10} & -2.50\times10^{-8} & -6.22\times10^{-8} \\ 4.24\times10^{-6} & -3.79\times10^{-5} & -5.84\times10^{-5} & -1.91\times10^{-3} & -3.65\times10^{-7} & -3.95\times10^{-5} & -1.57\times10^{-8} \\ -2.21\times10^{-7} & 1.99\times10^{-6} & 3.06e\times10^{-6} & 1.00\times10^{-4} & 1.91\times10^{-8} & 2.07\times10^{-6} & 1.07\times10^{-8} \\ 1.79\times10^{-9} & -1.60\times10^{-8} & -2.46\times10^{-8} & -8.07\times10^{-7} & -1.54\times10^{-10} & -1.67\times10^{-8} & 5.31\times10^{-11} \end{bmatrix}^T$$

$$P_s = 10^8 \times \begin{bmatrix} 2.5343 & -0.2893 & 0.0596 & 0.0056 & 1.0865 & 0.0113 & -0.1385 \\ -0.2893 & 2.3353 & -1.5603 & -0.0701 & -1.2734 & 0.0421 & -0.0069 \\ 0.0596 & -1.5603 & 3.0612 & 0.1503 & 3.0576 & -0.1907 & 0.0417 \\ 0.0056 & -0.0701 & 0.1503 & 0.0405 & 0.1650 & -0.0587 & -0.0005 \\ 1.0865 & -1.2734 & 3.0576 & 0.1650 & 5.7402 & 0.0968 & -0.0025 \\ 0.0113 & 0.0421 & -0.1907 & -0.0587 & 0.0968 & 2.9988 & -0.0008 \\ -0.1385 & -0.0069 & 0.0417 & -0.0005 & -0.0025 & -0.0008 & 0.0105 \end{bmatrix}$$

$$Q_s = 10^8 \times \begin{bmatrix} 0.0687 & -0.0253 & 0.0667 & 0.0098 & 0.1241 & 0.0018 & -0.0029 \\ -0.0253 & 0.3092 & -0.3552 & -0.0868 & -0.3736 & -0.0166 & -0.0006 \\ 0.0667 & -0.3552 & 0.7049 & 0.1409 & 0.8329 & -0.0052 & 0.0004 \\ 0.0098 & -0.0868 & 0.1409 & 0.0390 & 0.1578 & -0.0005 & -0.0009 \\ 0.1241 & -0.3736 & 0.8329 & 0.1578 & 1.0808 & 0.0022 & -0.0001 \\ 0.0018 & -0.0166 & -0.0052 & -0.0005 & 0.0022 & 0.1176 & -0.0000 \\ -0.0029 & -0.0006 & 0.0004 & -0.0009 & -0.0001 & -0.0000 & 0.0004 \end{bmatrix}$$

$$R_s = 10^4 \times \begin{bmatrix} 0.0932 & -0.0002 \\ -0.0002 & 1.6236 \end{bmatrix}$$

References

1. Rao, D.V.; Go, T.H. Automatic landing system design using sliding mode control. *Aerosp. Sci. Technol.* **2014**, *32*, 180–187.
2. Adibi, S.A.; Forer, S.; Fries, J.; Yliniemi, L. Autonomous Unmanned Aerial Vehicle (UAV) landing in windy conditions with MAP-Elites. *Knowl. Eng. Rev.* **2017**, *32*. [CrossRef]
3. Zhen, Z.; Jiang, S.; Ma, K. Automatic carrier landing control for Unmanned Aerial Vehicles based on preview control and particle filtering. *Aerosp. Sci. Technol.* **2018**, *81*, 99–107. [CrossRef]
4. Stevens, B.L.; Lewis, F.L.; Johnson, E.N. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
5. Sedlmair, N.; Theis, J.; Thielecke, F. Flight Testing Automatic Landing Control for Unmanned Aircraft Including Curved Approaches. *J. Guid. Control. Dyn.* **2022**, *45*, 726–739. [CrossRef]
6. Wang, L.; Jiang, X.; Zhang, Z.; Wen, Z. Lateral automatic landing guidance law based on risk-state model predictive control. *ISA Trans.* **2022**, *128*, 611–623. [CrossRef]
7. Yang, L.; Wang, C.; Wang, L. Autonomous UAVs landing site selection from point cloud in unknown environments. *ISA Trans.* **2022**, *130*, 610–628. [CrossRef]
8. Yuan, Y.; Duan, H.; Zeng, Z. Automatic Carrier Landing Control with External Disturbance and Input Constraint. *IEEE Trans. Aerosp. Electron. Syst.* **2022**. [CrossRef]
9. Mathisen, S.; Gryte, K.; Gros, S.; Johansen, T.A. Precision deep-stall landing of fixed-wing UAVs using nonlinear model predictive control. *J. Intell. Robot. Syst.* **2021**, *101*, 1–15. [CrossRef]
10. Qu, W.; Zhang, H.; Dong, Y. Optimization of UAV's Landing Longitudinal Control under Wind Disturbance. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *693*, 012106. [CrossRef]
11. Lungu, R.; Lungu, M. Application of H2/H-infinity and dynamic inversion techniques to aircraft landing control. *Aerosp. Sci. Technol.* **2015**, *46*, 146–158. [CrossRef]
12. Lungu, R.; Lungu, M.; Grigorie, L.T. Automatic control of aircraft in longitudinal plane during landing. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 1338–1350. [CrossRef]
13. Zhang, D.; Wang, X. Autonomous landing control of fixed-wing UAVs: From theory to field experiment. *J. Intell. Robot. Syst.* **2017**, *88*, 619–634. [CrossRef]
14. Kurnaz, S.; Çetin, O. Autonomous navigation and landing tasks for fixed wing small unmanned aerial vehicles. *Acta Polytech. Hung.* **2010**, *7*, 87–102.

15. Juang, J.G.; Yu, S.T. Disturbance encountered landing system design based on sliding mode control with evolutionary computation and cerebellar model articulation controller. *Appl. Math. Model.* **2015**, *39*, 5862–5881. [CrossRef]
16. Juang, J.G.; Cheng, C.J.; Yang, T.C. Wind shear encountered landing control based on CMACs. *Appl. Mech. Mater.* **2013**, *284*, 2351–2355. [CrossRef]
17. Di Cairano, S.; Bemporad, A. Model predictive control tuning by controller matching. *IEEE Trans. Autom. Control* **2009**, *55*, 185–190. [CrossRef]
18. Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Scokaert, P.O. Constrained model predictive control: Stability and optimality. *Automatica* **2000**, *36*, 789–814. [CrossRef]
19. Qin, S.J.; Badgwell, T.A. A survey of industrial model predictive control technology. *Control. Eng. Pract.* **2003**, *11*, 733–764. [CrossRef]
20. Hartley, E.N.; Maciejowski, J.M. Designing output-feedback predictive controllers by reverse-engineering existing LTI controllers. *IEEE Trans. Autom. Control* **2013**, *58*, 2934–2939. [CrossRef]
21. Tran, Q.N.; Özkan, L.; Backx, A. Generalized predictive control tuning by controller matching. *J. Process. Control* **2015**, *25*, 1–18. [CrossRef]
22. Glover, K.; McFarlane, D. Robust stabilization of normalized coprime factor plant descriptions with H-infinity bounded uncertainty. *IEEE Trans. Autom. Control* **1989**, *34*, 821–830. [CrossRef]
23. Tamkaya, K.; Ucun, L.; Ustoglu, I. H-infinity based model following method in autolanding systems. *Aerosp. Sci. Technol.* **2019**, *94*, 105379. [CrossRef]
24. Stevens, B.L.; Lewis, F.L. *Aircraft Control and Simulation*; John Wiley and Sons, Inc.: Hoboken, NY, USA, 1992.
25. Woodfield, A.A.; Woods, J.F. *Worldwide Experience of Wind Shear during 1981–1982*; Technical report; Royal Aircraft EstablIshment Bedford: Bedford, UK, 1983.
26. Bortoff, S.A.; Schwerdtner, P.; Danielson, C.; Di Cairano, S.; Burns, D.J. H-Infinity loop-shaped model predictive control with HVAC application. *IEEE Trans. Control. Syst. Technol.* **2022**, *30*, 2188–2203. [CrossRef]
27. Skogestad, S.; Postlethwaite, I. *Multivariable Feedback Control: Analysis and Design*; Wiley: New York, NY, USA, 2007; Volume 2.
28. McFarlane, D.; Glover, K. A loop-shaping design procedure using H-infinity synthesis. *IEEE Trans. Autom. Control* **1992**, *37*, 759–769. [CrossRef]
29. Latif, Z.; Shahzad, A.; Samar, R.; Bhatti, A.I. Lateral Parameter-Varying Modelling and Control of a UAV on-Ground. In Proceedings of the 4th IFAC Workshop on Linear Parameter Varying Systems LPVS 2021, Milan, Italy, 19–20 July 2021; pp. 130–135.
30. Hyde, R.A.; Glover, K. The application of scheduled H-infinity controllers to a VSTOL aircraft. *IEEE Trans. Autom. Control* **1993**, *38*, 1021–1039. [CrossRef]
31. Priess, M.C.; Conway, R.; Choi, J.; Popovich, J.M.; Radcliffe, C. Solutions to the inverse LQR problem with application to biological systems analysis. *IEEE Trans. Control. Syst. Technol* **2014**, *23*, 770–777. [CrossRef]
32. Jos, F. Sturm SeDuMi Home Page. Available online: https://sedumi.ie.lehigh.edu/ (accessed on 1 November 2022).
33. Johan Löfberg YALMIP Home Page. Available online: http://yalmip.github.io/ (accessed on 1 November 2022).
34. Jerez, J.L.; Kerrigan, E.C.; Constantinides, G.A. A condensed and sparse QP formulation for predictive control. In Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, 12–15 December 2011; pp. 5217–5222.