DP2: Flying Wing

Daniel Klos[*] and Luke Brown[†]
Undergraduate Students at the University of Illinois Urbana-Champaign, Urbana, Illinois, 61801

This report presents the design, implementation, and testing of an autopilot system for a Zagi-style flying wing, developed to achieve controlled landings on a floating runway. The aircraft's tailless design introduces stability challenges, requiring a robust controller for reliable descent. A nonlinear model is developed, linearized at a trim state, and analyzed for controllability. A Linear Quadratic Regulator (LQR) is designed, iterating on Q and R weight matrices to maximize landing success. Simulations with randomized initial conditions identified the best-performing weights, achieving a INSERT RATE HERE% landing success rate. A video summarizing our methods, results, and stakeholder impact can be found in the references section [1].

I. Nomenclature

p_x, p_y, p_z = x, y and z components of glider position expressed in the world frame (m)
ψ, θ, ϕ = yaw, pitch and roll (rad)
v_x, v_y, v_z = x, y and z components of linear velocity of the glider expressed in the body frame (m/s)
w_x, w_y, w_z = x, y and z components of angular velocity of the glider expressed in the body frame (rad/s)
δ_r, δ_l = right and left elevon deflection angles (rad)
m = mass of the glider (kg)
J^B = moment of inertia matrix for the glider expressed in the body frame (kg $\cdot$ m^2)
R_B^W = rotation matrix that describes the coordinates of the body frame in world frame coordinates
N = transformation matrix that converts angular velocity to angular rates
f_x, f_y, f_z = aerodynamic forces in the x, y and z directions (N)
τ_x, τ_y, τ_z = aerodynamic torque about the x, y and z axes (N $\cdot$ m)
f, f_{red} = coefficient matrix and its reduced form ($\dot{\zeta} = f_{\text{red}}(\zeta_{\text{red}}, \mu)$)
K = gain matrix for linear state feedback (2 x 10)
$\zeta, \zeta_{\text{red}}, \mu$ = state, reduced state and input of nonlinear system
x, u = state and input of linear system
A, B = matrices that describe the state-space model
C = controllability matrix $\left(C = \begin{bmatrix} B & AB & A^2B & \dots & A^9B \end{bmatrix}\right)$
Q_i, R_i = iterated weight matrices used to tune LQR (for i = 0, 1, ... 8)
q_α, r_δ = weight matrix entries for each reduced state variable (α)

II. Introduction

This project aims to develop an autopilot system that ensures a safe landing on a floating runway.

The approach begins with modeling the aircraft's dynamics and identifying a suitable trim condition for controlled descent. The system is then linearized and analyzed for controllability, followed by the design and implementation of a feedback controller. Performance is evaluated through simulations with randomized initial conditions, with a requirement of at least an 90% safe landing success rate.

By refining control strategies for tailless aircraft, this project contributes to the broader field of autonomous flight and adaptive control systems.

[*] Undergraduate, Aerospace Engineering
[†] Undergraduate, Aerospace Engineering

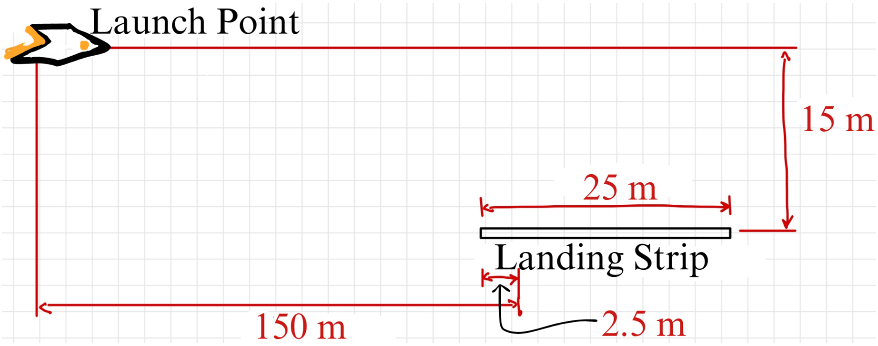

Fig. 1 Schematic of the launch position of the Zagi-style aircraft, relative to the landing strip [2].

III. Theory

A. Equations of Motion

The dynamics of the glider are modeled by nonlinear equations that capture its behavior under various forces and torques [2]:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix} = R_B^W(\psi, \theta, \phi) \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \qquad \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = N(\psi, \theta, \phi) \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \frac{1}{m} \left(R_B^W(\psi, \theta, \phi) \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} f_x(v_x, v_y, v_z, w_x, w_y, w_z, \delta_r, \delta_l) \\ f_y(v_x, v_y, v_z, w_x, w_y, w_z, \delta_r, \delta_l) \\ f_z(v_x, v_y, v_z, w_x, w_y, w_z, \delta_r, \delta_l) \end{bmatrix} - \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \times m \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \right) \tag{2}$$

$$\begin{bmatrix} \dot{w}_x \\ \dot{w}_y \\ \dot{w}_z \end{bmatrix} = J^{B^{-1}} \left(\begin{bmatrix} \tau_x(v_x, v_y, v_z, w_x, w_y, w_z, \delta_r, \delta_l) \\ \tau_y(v_x, v_y, v_z, w_x, w_y, w_z, \delta_r, \delta_l) \\ \tau_z(v_x, v_y, v_z, w_x, w_y, w_z, \delta_r, \delta_l) \end{bmatrix} - \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \times J^B \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \right). \tag{3}$$

The three-dimensional position values representing the location of our glider are denoted as p_x, p_y, p_z and their corresponding velocities as v_x, v_y, v_z. The yaw, pitch and roll angles are represented as ψ, θ, ϕ, respectively, with their angular velocities as w_x, w_y, w_z.

To express this system in standard form, we introduce the state (ζ) and input (μ) vectors of the nonlinear system:

$$\zeta = \begin{bmatrix} p_x & p_y & p_z & \psi & \theta & \phi & v_x & v_y & v_z & w_x & w_y & w_z \end{bmatrix}^\top, \quad \mu = \begin{bmatrix} \delta_r & \delta_l \end{bmatrix}^\top. \tag{4}$$

We can now rewrite the equations in the form:

$$\dot{\zeta} = f(\zeta, \mu), \tag{5}$$

where $f(\zeta, \mu)$ is defined by the right-hand side of Eqs. (1)–(3).

B. Equilibrium Points

To linearize the system in Eq. 5, we first need to find a valid equilibrium point (trim condition). To do this, we needed to find a steady-state solution where the time derivatives of the state variables are zero.

The full system consists of 12 state variables, but not all are necessary for determining trim conditions. In particular, the position coordinates that need to change in order for a steady, straight descent p_x, p_z are excluded, as they do not directly influence the trim dynamics.

Thus, we define the reduced state vector:

$$\zeta_{\text{red}} = \begin{bmatrix} p_y & \psi & \theta & \phi & v_x & v_y & v_z & w_x & w_y & w_z \end{bmatrix}^\top, \tag{6}$$

With this reduction, the equations of motion now take the form:

$$\dot{\zeta_{\text{red}}} = f_{\text{red}}(\zeta_{\text{red}}, \mu), \tag{7}$$

where f_{red} is obtained by selecting the relevant equations from the full nonlinear model (f).

We numerically solved for equilibrium values by minimizing the squared norm of these reduced equations [3]. This approach finds state and control values that satisfy the steady-state condition:

$$\|f_{\text{red}}(\zeta_{\text{red}}, \mu)\|^2 = 0. \tag{8}$$

The equilibrium points obtained from the minimization process are:

$$\zeta_{\text{e}} = \begin{bmatrix} 0 & 0 & 0.128 & 0 & 6.709 & 0 & 1.438 & 0 & 0 & 0 \end{bmatrix}^\top, \quad \mu_e = \begin{bmatrix} -0.419 & -0.419 \end{bmatrix}^\top \tag{9}$$

To verify that these values correspond to an equilibrium state, they were substituted back into the reduced equations. The computed values of $f_{\text{red}}(\zeta_{\text{e}}, \mu_e)$ were on the order of 10^{-6}, confirming that the selected state is close to a true equilibrium.

C. Linearization

Since the equations of motion are nonlinear, we linearized them around the equilibrium point to obtain a state-space representation suitable for control design. This involved introducing the linear state vector (x) and the linear state input vector (μ).

$$x = \zeta_{\text{red}} - \zeta_{\text{e}}, \quad u = \mu - \mu_{\text{e}} \tag{10}$$

Now Eq. 5 can be approximated close to the chosen equilibrium point by the linear system

$$\dot{x} = Ax + Bu \tag{11}$$

where

$$A = \left.\frac{\partial f_{\text{red}}}{\partial x}\right|_{\zeta_{\text{e}}, \mu_e}, \quad B = \left.\frac{\partial f_{\text{red}}}{\partial u}\right|_{\zeta_{\text{e}}, \mu_e}. \tag{12}$$

In order to verify the controllability of our approximated linear system, we must confirm that the controllability matrix C is full rank [5].

$$C = \begin{bmatrix} B & AB & A^2B & \dots & A^9B \end{bmatrix}. \tag{13}$$

For our system, we compute C and verify its rank using numerical methods. The computed rank of C is:

$$\text{rank}(C) = 10, \tag{14}$$

which matches the number of states in x, confirming that the system is fully controllable. Therefore, state feedback control can be successfully applied.

D. Design Linear State Feedback

For our system, we chose to apply the input of the form

$$u = -Kx, \tag{15}$$

where K is a 2x10 matrix with entries k_{ij} representing the respective choice of gains.

In order to systematically determine the optimal gain matrix, we applied the Linear Quadratic Regulator (LQR) approach [4] to our system. LQR minimizes a quadratic cost function over an infinite time horizon, balancing the trade-off between state deviations and control effort. The initial weight matrices used were:

$$Q_0 = \mathrm{diag}(q_{p_y}, q_\psi, q_\theta, q_\phi, q_{v_x}, q_{v_y}, q_{v_z}, q_{w_x}, q_{w_y}, q_{w_z}), \quad R_0 = \mathrm{diag}(r_\delta, r_\delta) \tag{16}$$

$$Q_0 = \mathrm{diag}(1, 1, 10, 2, 1, 5, 10, 2, 10, 2), \quad R_0 = \mathrm{diag}(10, 10) \tag{17}$$

The weight matrices Q and R prioritize control efforts based on the importance of each state in achieving a stable landing.

- **Position and Velocity:** q_{p_y} and q_{v_x} have low weights, as lateral deviation and forward speed are less critical. Higher weights on q_θ and q_{v_z} ensure controlled descent, while q_{v_y} is moderately weighted to reduce drift.
- **Orientation and Angular Velocity:** q_ψ and q_ϕ have lower weights, as they mainly aid stability. q_θ and q_{w_y} are strongly weighted to maintain descent control, while q_{w_x} and q_{w_z} are given moderate weights to prevent excessive oscillations.

The control effort weight r_δ penalizes large elevon deflections to avoid aggressive, unstable corrections while maintaining smooth actuation.

Using this method, we obtained our gain matrix K to be:

$$K = \begin{bmatrix} -0.2236 & -4.697 & -1.693 & -3.576 & 0.1039 & -0.7434 & -0.008926 & -0.3687 & -0.6613 & -3.992 \\ 0.2236 & 4.697 & -1.693 & 3.576 & 0.1039 & 0.7434 & -0.008926 & 0.3687 & -0.6613 & 3.992 \end{bmatrix} \tag{18}$$

After calculating our gain matrix, we checked K's asymptotic stability by calculating the eigenvalues (s_i) of $A - BK$:

$$\lambda(A - BK) = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\}, \tag{19}$$

where $s_1 = -12.58$, $s_2 = -13.64 + 1.695i$, $s_3 = -13.64 - 1.695i$, $s_4 = -2.721 + 2.903i$, $s_5 = -2.721 - 2.903i$, $s_6 = -0.2089 + 0.5171i$, $s_7 = -0.2089 - 0.5171i$, $s_8 = -0.7533$, $s_9 = -0.8374 + 1.463i$, and $s_{10} = -0.8374 - 1.463i$. All of s_i have negative real part, confirming the stability of our closed loop system.

IV. Experimental Methods

We decided to define success for our controller as being able to land the glider at least 90 percent of the time, where ψ_0, θ_0 and ϕ_0 were randomly sampled in the range of $[-\pi/6, \pi/6]$ rad, and v_x in $[2.5, 5.5]$ m/s, with all others set to zero. To fine-tune the controller and aim for this target performance, we systematically varied the weighting matrices Q and R in our cost function and evaluated their impact on landing success. For each Q, R pair, we ran 100 simulations, recording the landing rate in Table 1. To ensure robustness, we recorded the initial conditions of each failed attempt, allowing us to effectively tune the weights of our Q and R matrices.

We started with our initial Q and R weight matrices and ran 100 simulations. We then systematically tweaked the weights of Q and R based on the initial conditions of the failed tests.

INCLUDE JUSTIFICATION FOR TUNING HERE

Table 1 Comparison of Q and R matrices and their corresponding success rates

Q, R	r_δ	Success Rate (%)	Q Entries									
			q_{p_y}	q_ψ	q_θ	q_ϕ	q_{v_x}	q_{v_y}	q_{v_z}	q_{w_x}	q_{w_y}	q_{w_z}
Q_0, R_0	10	97	1	1	10	2	1	5	10	2	10	2
Q_1, R_1	12	99	1	1	10	2	1	5	10	2	10	2
Q_2, R_2	12	94	2	1	10	2	1	5	10	2	10	2
Q_3, R_3	12	91	3	1	10	2	1	5	10	2	10	2
Q_4, R_4	12	96	1	1	10	2	1	6	10	2	10	2
Q_5, R_5	12	97	1	1	10	2	1	7	10	2	10	2
Q_6, R_6	12	95	1	1	10	2	1	8	10	2	10	2
Q_7, R_7	10	97	1	1	10	2	1	7	10	2	10	2
Q_8, R_8	9	93	1	1	10	2	1	7	10	2	10	2

V. Results and Discussion

To assess the robustness of our control design, we expanded from 100 to 500 simulations using the best-performing weight matrix (Q_1, R_1), which initially achieved a 99% success rate. This larger dataset provides a more statistically significant performance measure and helps define failure boundaries. Simulations covered a range of initial conditions from Section IV, ensuring a thorough stress test. Our goal was to verify that success consistently exceeds our 90% requirement, while identifying failure cases.

The results, summarized in Table 2, confirm that the controller reliably handles nominal conditions and reveal a specific range of initial conditions where it achieved a 100% success rate. Beyond this range, failures were primarily linked to extreme attitudes or velocity perturbations, highlighting areas for potential refinement.

Table 2 Range of initial conditions where the controller achieved 100% success.

Parameter	Min Value	Max Value
v_x (m/s)	4.8	5.2
ψ (rad)	-$\pi/12$	$\pi/12$
θ (rad)	-$\pi/12$	$\pi/12$
ϕ (rad)	-$\pi/12$	$\pi/12$

VI. Conclusion

This study developed and tested an autopilot system for a Zagi-style flying wing, demonstrating its effectiveness in achieving safe landings on a floating runway. Through systematic tuning of the LQR weight matrices, we identified the best-performing configuration (Q_1, R_1) among the nine iterations tested, achieving a success rate of 95.4 % over 500 simulations, exceeding the 90% requirement. The results highlight the controller's robustness under nominal conditions and define a range of initial states where it ensures 100% success.

Future work could refine the control system by incorporating adaptive or nonlinear control techniques to handle more extreme initial conditions. Additionally, incorporating real-world disturbances, such as wind gusts or sensor noise, would provide a more rigorous validation of the system's reliability. Extending this framework to hardware implementation would further bridge the gap between simulation and practical application.

Appendix

Cat Pilots' Review

Our cat pilots provided valuable feedback on the landing system's safety, comfort, and reliability. Their experiences highlight the controller's strengths and areas for refinement.

Pilot Testimonies:

Pilot 1: "The descent was a bit shaky, but the controller recovered well, and I landed safely. A smoother transition to final approach would be nice."

Pilot 2: "I started with a high roll angle and thought I was doomed, but the controller corrected just in time. Impressive work!"

Pilot 3: "I started out way too fast with a negative pitch—I knew I wasn't making it. I missed the runway by a few meters and took an unexpected swim."

Key Concerns and Suggestions:

- **Stability on Approach:** Some pilots experienced oscillations during descent. Refining damping control could improve landing consistency.
- **Handling Extreme Conditions:** The controller successfully corrected aggressive roll angles but struggled with excessive velocity and negative pitch.
- **Landing Margin:** Small miscalculations at high speeds led to runway overshoots. Further tuning could improve recovery in edge cases.

Overall, the controller performed reliably within a reasonable range of initial conditions, but extreme velocities and pitch angles remain a challenge. Future improvements should enhance robustness under these conditions.

Acknowledgments

We sincerely thank Professor Timothy Bretl for designing this engaging project, as well as the TAs and CAs for their guidance and technical support. Their feedback and office hours were invaluable in refining our controller and report.

References

[1] Klos, D., Brown, L., "DP2: Flying Wing Video", online video, 2025.

[2] Professor Bretl, "Design Project 2 (flying wing)", AE 353: Aerospace Control Systems, University of Illinois, Spring 2025.

[3] SciPy Developers, "scipy.optimize.minimize," SciPy Documentation, retirieved 28 Febuary 2025. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html`

[4] Professor Bretl, "Optimazation and Optimal Control", AE 353: Aerospace Control Systems, University of Illinois, Spring 2025.

[5] Professor Bretl, "Eigenvalue Placement", AE 353: Aerospace Control Systems, University of Illinois, Spring 2025.