Robust H_∞ Pitch Control Applied to a Fixed-Wing Unmanned Aerial Vehicle

ECE 854: Robust Control
Michigan State University
Spring 2026

Julio Quiroga Galan

April 30, 2026

Abstract

This project studies the design and evaluation of a robust H_∞ controller for the pitch-attitude dynamics of a fixed-wing unmanned aerial vehicle (UAV). The work is based on the paper "Robust H_∞ Control Applied on a Fixed Wing Unmanned Aerial Vehicle," where robust longitudinal and lateral controllers are developed for a quarter-scale Piper J3-Cub UAV. In this project, the focus is restricted to the longitudinal elevator-to-pitch channel to reproduce the main robust control ideas in a clear and organized way. A linear longitudinal model is built from the state-space matrices reported in the paper, and parametric uncertainty is introduced for selected aerodynamic derivatives and control-effectiveness terms. A mixed-sensitivity H_∞ controller is then synthesized in MATLAB using weighting functions motivated by the paper. Simulation results show that the nominal closed-loop system is stable and achieves fast pitch tracking with no overshoot. Uncertain closed-loop simulations also remain stable across all sampled plants. Robust-stability analysis confirms that the controller preserves stability for the modeled uncertainty, although robust-performance analysis shows that performance is not guaranteed for the entire uncertainty set. Overall, the controller achieves strong robust stability and good nominal performance, but only limited robust performance under the selected uncertainty description.

1 Introduction

Fixed-wing unmanned aerial vehicles are widely used in surveillance, inspection, environmental monitoring, and defense-related applications. Because these systems operate under varying flight conditions and are affected by modeling errors, disturbances, and sensor noise, classical fixed-gain control methods may experience degraded performance when the actual plant departs from its nominal model. For this reason, robust control methods are attractive for UAV applications.

One important robust control framework is H_∞ control, which allows the designer to shape closed-loop performance while explicitly accounting for uncertainty and disturbances. In the selected reference paper, robust H_∞ controllers are developed for pitch, roll, and yaw autopilots of a fixed-wing UAV using a μ-synthesis-based design framework. The results in the paper show that robust control improves stability and tracking performance under perturbations, disturbances, and noise.

The purpose of this project is to reproduce and study part of that work in MATLAB. Instead of implementing the full longitudinal and lateral design, this project focuses only on the pitch-attitude control problem. This narrower scope allows the main modeling assumptions, uncertainty description, controller design steps, and simulation results to be presented clearly while still meeting the objective of the course project.

2 Problem Formulation

The goal of this project is to design a robust controller for the longitudinal pitch dynamics of a fixed-wing UAV. More specifically, the control input is the elevator deflection and the controlled output is the pitch angle. The design objectives are:

- stabilize the nominal longitudinal plant,
- achieve fast and well-damped pitch-angle tracking,
- maintain acceptable closed-loop behavior under plant uncertainty,
- reduce sensitivity to measurement noise and modeling errors.

The selected paper develops robust controllers for the full UAV attitude dynamics, but in this project only the pitch channel is considered. This corresponds to the elevator-to-pitch subsystem extracted from the longitudinal model. The problem is therefore treated as a single-input single-output robust control problem based on the reported linearized longitudinal dynamics.

3 Modeling and Uncertainty Description

The paper provides the linear longitudinal state-space model of the UAV around a trim condition. The state vector is taken as

$$x = \begin{bmatrix} \Delta u & \Delta w & \Delta q & \Delta\theta \end{bmatrix}^T,$$

where Δu and Δw are perturbations in the body-axis velocity components, Δq is the pitch-rate perturbation, and $\Delta\theta$ is the pitch-angle perturbation.

The nominal longitudinal model used in this project is

$$\dot{x} = Ax + Bu, \qquad y = Cx + Du \tag{1}$$

with

$$A = \begin{bmatrix} -0.06729 & 1.052 & 0 & -9.806 \\ -1.346 & -8.613 & 15 & 0 \\ 0.4848 & -8.606 & -20.95 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ -11.43 \\ -107.5 \\ 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}, \qquad D = 0.$$

The system input u is the elevator deflection δ_e, and the output y is the pitch angle θ.

To account for model uncertainty, selected longitudinal derivatives were modeled as uncertain real parameters. Specifically, uncertainty was introduced in aerodynamic and control-effectiveness terms associated with the entries of the A and B matrices. A 20% uncertainty level was applied to the following nominal parameters:

$$X_u,\ X_w,\ Z_u,\ Z_w,\ M_u,\ M_w,\ M_q,\ Z_{\delta_e},\ M_{\delta_e}.$$

This uncertainty model does not reproduce the exact uncertainty structure of the original paper, but it provides a reasonable parametric uncertainty description suitable for evaluating robust stability and robust performance in MATLAB. The uncertain plant therefore represents a family of longitudinal pitch-channel models around the nominal trim condition.

4 Robust Control Method

To design the controller, a mixed-sensitivity H_∞ formulation was used. The objective is to find a controller $K(s)$ that minimizes

$$\left\| \begin{bmatrix} W_1 S \\ W_2 K S \\ W_3 T \end{bmatrix} \right\|_\infty,$$

where

$$S = \frac{1}{1+GK}, \qquad T = \frac{GK}{1+GK}.$$

Here, S is the sensitivity function and T is the complementary sensitivity function. The weighting functions were chosen based on the pitch-control case reported in the paper:

$$W_1(s) = \frac{0.1s + 5.973}{s + 0.176},$$

$$W_2(s) = \frac{0.8s + 42.91}{s + 119.2},$$

$$W_3(s) = 0.01.$$

The interpretation of these weights is as follows:

- W_1 penalizes tracking error and shapes the low-frequency performance,
- W_2 penalizes excessive control effort,
- W_3 limits high-frequency amplification and helps reduce noise sensitivity.

The controller was synthesized in MATLAB using the `mixsyn` command. This yields an H_∞ controller that balances tracking performance, control effort, and robustness.

The frequency-domain behavior of the nominal closed-loop transfer function $T(s)$ is shown in Fig. 1. The plot indicates that the closed-loop response has strong attenuation at high frequencies, which is desirable for limiting the effect of high-frequency uncertainty and measurement noise.

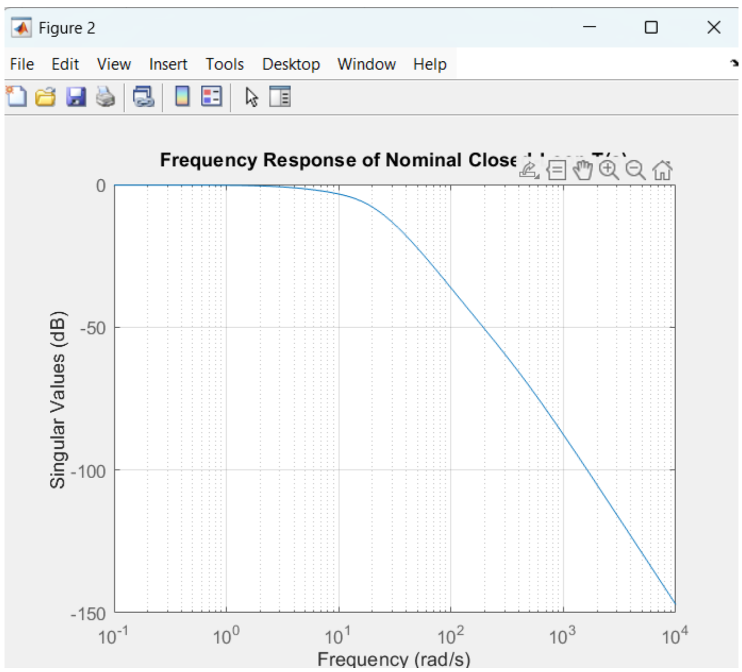

Figure 1: Frequency response of the nominal closed-loop transfer function $T(s)$.

The tradeoff between disturbance rejection and bandwidth can be examined more directly through the sensitivity and complementary sensitivity functions. Fig. 2 shows that $S(s)$ is small at low frequencies, which supports good tracking and disturbance rejection, while $T(s)$ rolls off at high frequencies, which helps attenuate noise and unmodeled dynamics.

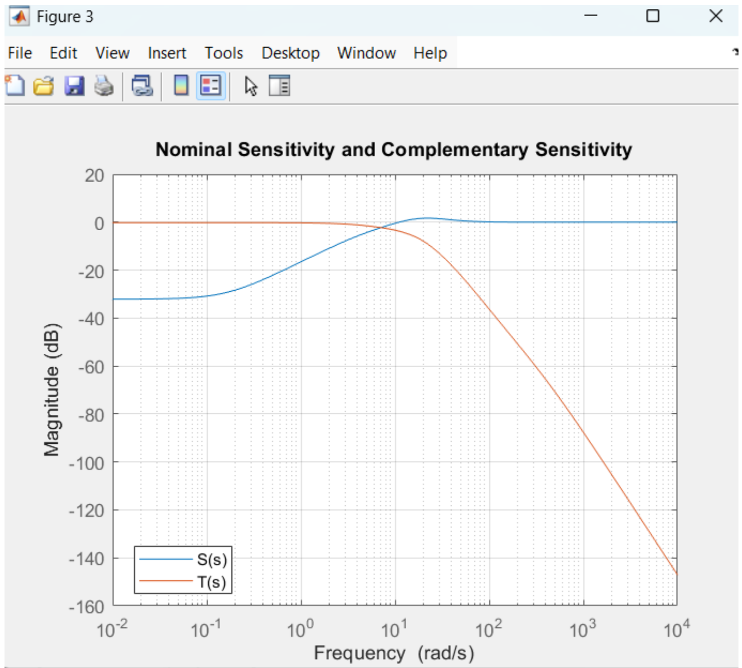

Figure 2: Nominal sensitivity and complementary sensitivity functions.

5 Results

The synthesized controller achieved a mixed-sensitivity value of

$$\gamma = 0.8784.$$

The nominal closed-loop poles were located in the open left-half plane, which confirms nominal closed-loop stability. The nominal step response is shown in Fig. 3. The response is fast, well-damped, and has no overshoot. The main time-domain characteristics were:

- rise time: 0.2398 s,
- settling time: 0.5776 s,
- overshoot: 0%,
- peak value: 0.9749.

These values indicate that the nominal controller produces fast tracking with no overshoot and good damping.

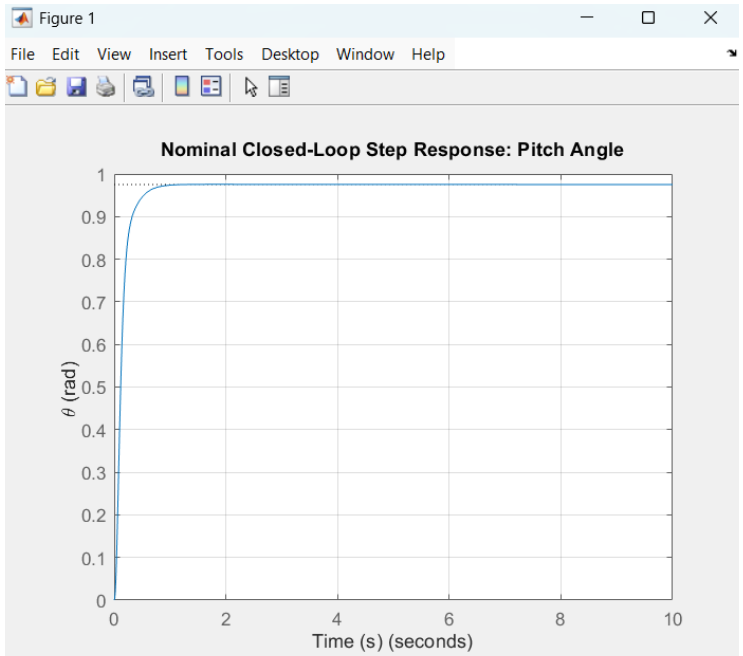

Figure 3: Nominal closed-loop step response for the pitch angle.

The sensitivity norms were also computed as

$$\|S\|_\infty = 1.2153, \qquad \|T\|_\infty = 0.9752.$$

These results indicate acceptable closed-loop robustness and limited amplification of the complementary sensitivity response.

To evaluate uncertain performance, twelve samples of the uncertain plant were generated and closed with the same controller. The corresponding responses are shown in Fig. 4. All twelve uncertain closed-loop systems remained stable, and their trajectories remained close to the nominal response. This indicates that the controller provides good robustness with respect to the modeled parameter variations.

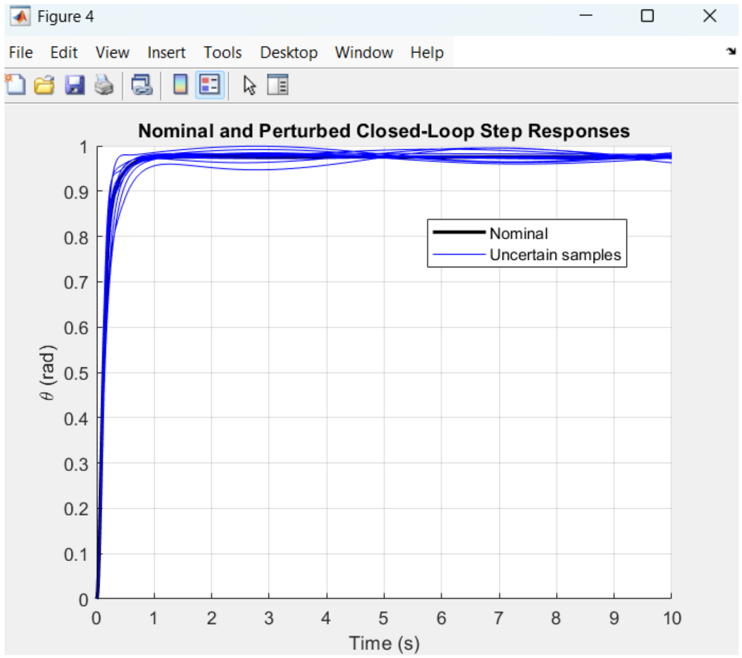

Figure 4: Nominal and perturbed closed-loop step responses for the pitch angle.

The corresponding control effort for a unit step pitch reference is shown in Fig. 5. The elevator command exhibits a strong initial transient and then decays smoothly toward its steady value. This confirms that the controller achieves good tracking without requiring unbounded actuator action.

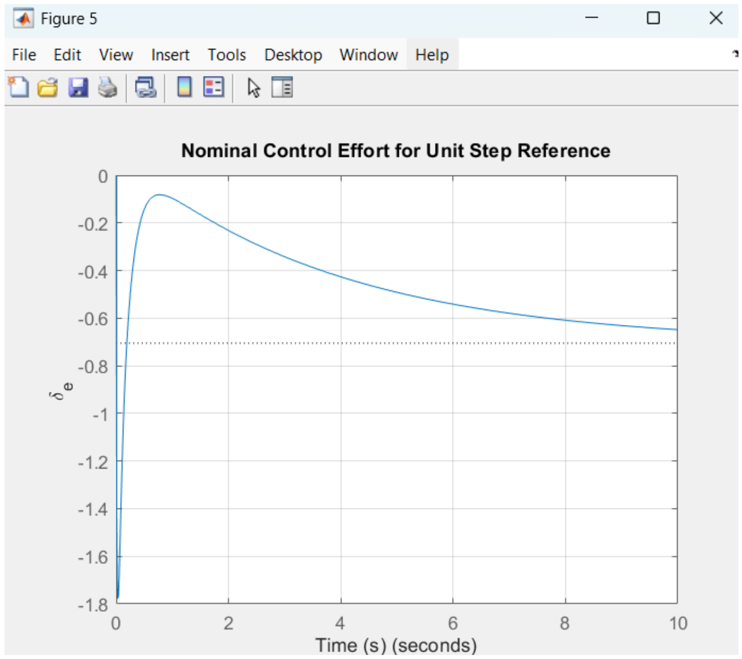

Figure 5: Nominal control effort for a unit step pitch-angle reference.

The relationship between reference tracking and measurement-noise transmission is shown in Fig. 6. At low frequencies, the closed loop preserves good reference tracking, while at high frequencies the transmission rolls off strongly, indicating strong attenuation of measurement noise.

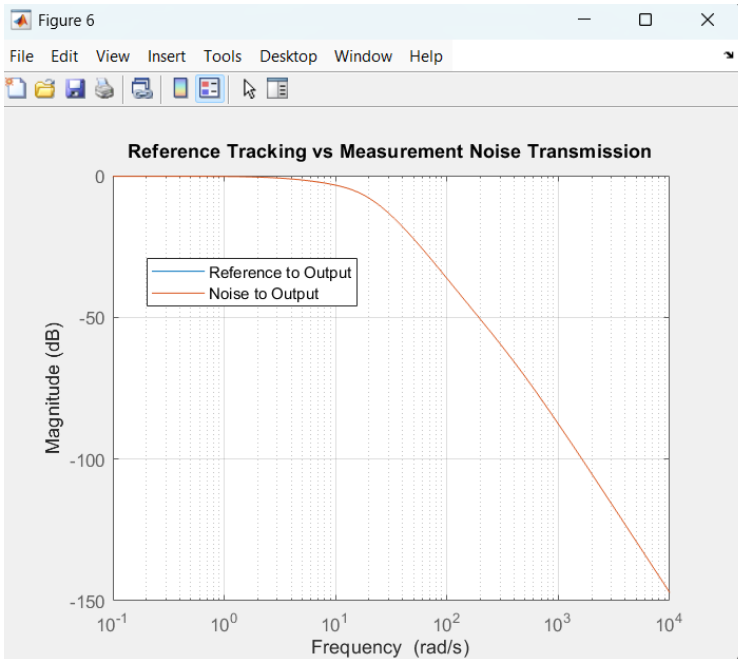

Figure 6: Reference tracking versus measurement-noise transmission.

A more formal robust-stability analysis was performed using `robuststab`. The result showed that the system is robustly stable for the modeled uncertainty and can tolerate approximately 196% of the uncertainty set before losing stability. The destabilizing perturbation was predicted near 0.682 rad/s. This is a strong indication that the controller provides a significant stability margin.

However, robust-performance analysis using `robustperf` showed that the uncertain system does not satisfy the desired performance objectives over the full uncertainty set. The result indicated that performance robustness is lost at about 79% of the modeled uncertainty. In other words, the system remains stable, but the closed-loop gain and tracking performance are not guaranteed to satisfy the chosen bounds for all uncertain plants.

6 Discussion

The results show a clear difference between robust stability and robust performance. From a stability point of view, the controller performs very well. The nominal plant is stable, all sampled uncertain plants are stable, and the robust-stability margin is greater than one. This means the closed-loop system preserves stability even when the uncertainty is significantly larger than the modeled level.

From a performance point of view, the result is more limited. While the nominal and sampled step responses shown in Figs. 3 and 4 appear satisfactory, the formal robust-performance test indicates that the selected weighting functions and controller do not guarantee the desired performance over the entire uncertainty set. Therefore, there is a tradeoff between conservatism, uncertainty size, and achievable closed-loop performance.

The frequency-domain plots in Figs. 1, 2, and 6 are consistent with this interpretation. The controller provides good low-frequency tracking and strong high-frequency attenuation, but these advantages do not fully guarantee robust performance for every uncertain plant in the modeled family.

One limitation of this project is that the uncertainty model was simplified to parametric perturbations in selected state-space entries. Also, only the pitch channel was studied instead of the full longitudinal and lateral UAV dynamics considered in the original paper. Nevertheless, the project successfully reproduces the main robust control ideas of the reference work and demonstrates how MATLAB robust control tools can be used to evaluate both stability and performance.

7 Conclusion

This project investigated robust H_∞ pitch control for a fixed-wing UAV based on the paper "Robust H_∞ Control Applied on a Fixed Wing Unmanned Aerial Vehicle." A longitudinal elevator-to-pitch model was implemented, parametric uncertainty was introduced, and a mixed-sensitivity H_∞ controller was synthesized in MATLAB.

The results showed that the controller achieves nominal closed-loop stability, fast pitch tracking, and no overshoot. Uncertain simulations further showed that the controller maintains stable behavior across all sampled plants. Robust stability analysis confirmed that the system remains stable for the modeled uncertainty with a substantial stability margin. However, robust performance analysis showed that performance objectives are not guaranteed for the entire uncertainty set.

Overall, the controller can be considered successful in achieving robust stability and good nominal behavior, but only partial robust performance. This highlights an important lesson in robust control: a system may remain stable under uncertainty even when its performance specifications are no longer fully satisfied.

Appendix: MATLAB Code

```
%% ECE 854 Final Project
% Option 2
%
% Julio Quiroga Galan
%
```

```
% Paper:
% "Robust H-infinity Control Applied on a Fixed Wing Unmanned Aerial Vehicle."
%
% Toolboxes required:
%   - Control System Toolbox
%   - Robust Control Toolbox

clear; clc; close all;

%% ----------------------------------------------------------
% 1) Nominal longitudinal model from the paper
% States: x = [Delta u; Delta w; Delta q; Delta theta]
% Input used: elevator deflection delta_e
% Output used: pitch angle theta
%
% The model and weights are stated in the selected paper.

A_nom = [ -0.06729   1.052    0       -9.806;
          -1.346    -8.613   15        0;
           0.4848   -8.606  -20.95     0;
           0         0       1         0 ];

B_nom_full = [   0      3.6;
               -11.43   0;
              -107.5    0;
                 0      0 ];

B_elev = B_nom_full(:,1);
C_theta = [0 0 0 1];
D_theta = 0;

G_nom = ss(A_nom, B_elev, C_theta, D_theta);

disp('Nominal elevator-to-pitch plant:');
G_nom

%% ----------------------------------------------------------
% 2) Uncertainty description
% The assignment requires an uncertain model or uncertainty description.
% Here we use uncertainty in selected entries of A and B.
%
% This is a reasonable course-project uncertainty model around the nominal
% plant and is consistent with the paper's robust control motivation under
% uncertainty, disturbances, and sensor noise.

Xu  = ureal('Xu',  -0.06729, 'Percentage', 20);
```

```
Xw   = ureal('Xw',    1.052,    'Percentage', 20);

Zu   = ureal('Zu',   -1.346,    'Percentage', 20);
Zw   = ureal('Zw',   -8.613,    'Percentage', 20);

Mu   = ureal('Mu',    0.4848,   'Percentage', 20);
Mw   = ureal('Mw',   -8.606,    'Percentage', 20);
Mq   = ureal('Mq',  -20.95,     'Percentage', 20);

Zde  = ureal('Zde', -11.43,     'Percentage', 20);
Mde  = ureal('Mde',-107.5,      'Percentage', 20);

A_unc = [ Xu     Xw     0     -9.806;
          Zu     Zw    15       0;
          Mu     Mw    Mq       0;
           0      0     1       0 ];

B_unc = [ 0;
          Zde;
          Mde;
          0 ];

G_unc = ss(A_unc, B_unc, C_theta, D_theta);

disp('Uncertain plant created.');

%% ------------------------------------------------------------
% 3) Weighting functions
% From Table 4 in the paper for the pitch case:
% Perf_theta = (0.1 s + 5.973)/(s + 0.176)
% W_elev     = (0.8 s + 42.91)/(s + 119.2)
% N_theta    = 0.01
%
% These are used here as mixed-sensitivity weights.

s = tf('s');

W1 = (0.1*s + 5.973) / (s + 0.176);      % performance weight
W2 = (0.8*s + 42.91) / (s + 119.2);      % control effort weight
W3 = 0.01;                               % noise / complementary sensitivity weight

disp('Weights defined.');

%% ------------------------------------------------------------
% 4) H-infinity mixed-sensitivity design
% The assignment requires a robust control or robustness analysis method.

```

```
try
    [K, CL, gamma_opt] = mixsyn(G_nom, W1, W2, W3);
catch ME
    error(['mixsyn failed. Check that Robust Control Toolbox is installed. Error: ', ME.message]);
end

disp('Controller K obtained.');
K
fprintf('Achieved gamma = %.4f\n', gamma_opt);

%% ---------------------------------------------------------
% 5) Closed-loop systems
L_nom = minreal(G_nom*K);
S_nom = minreal(feedback(1, L_nom));      % S = 1/(1+L)
T_nom = minreal(feedback(L_nom, 1));      % T = L/(1+L)
Tr_nom = T_nom;                           % reference to output

disp('Nominal closed-loop poles:');
disp(pole(Tr_nom));

if isstable(Tr_nom)
    disp('Nominal closed-loop is stable.');
else
    disp('Warning: nominal closed-loop is NOT stable.');
end

%% ---------------------------------------------------------
% 6) Basic norms
try
    gS = norm(S_nom, inf);
    gT = norm(T_nom, inf);
    fprintf('||S||_inf = %.4f\n', gS);
    fprintf('||T||_inf = %.4f\n', gT);
catch
    disp('Could not compute H-infinity norms exactly.');
end

%% ---------------------------------------------------------
% 7) Time vector
t = 0:0.01:10;

%% ---------------------------------------------------------
% 8) Nominal step response
figure;
step(Tr_nom, t);
grid on;
title('Nominal Closed-Loop Step Response: Pitch Angle');
xlabel('Time (s)');
```

```matlab
ylabel('\theta (rad)');

try
    info_nom = stepinfo(Tr_nom);
    disp('Nominal step response info:');
    disp(info_nom);
catch
    disp('stepinfo could not be computed.');
end

%% ---------------------------------------------------------
% 9) Singular value / magnitude plot
% sigma may not always be ideal for SISO if you want a simple plot, but it runs.
figure;
sigma(T_nom);
grid on;
title('Frequency Response of Nominal Closed-Loop T(s)');

%% ---------------------------------------------------------
% 10) Bode magnitude of sensitivity functions
figure;
bodemag(S_nom, T_nom);
grid on;
legend('S(s)','T(s)','Location','best');
title('Nominal Sensitivity and Complementary Sensitivity');

%% ---------------------------------------------------------
% 11) Perturbed / uncertain closed-loop step responses
N = 12;
stable_count = 0;
infos_unc = cell(N,1);

try
    G_samples = usample(G_unc, N);
catch ME
    error(['usample failed. Error: ', ME.message]);
end

figure;
hold on;
grid on;

% Plot nominal first
[y_nom, t_nom] = step(Tr_nom, t);
plot(t_nom, squeeze(y_nom), 'k', 'LineWidth', 2);

for k = 1:N
    Gk = G_samples(:,:,k);
    Tk = minreal(feedback(Gk*K, 1));
```

```
    if isstable(Tk)
        stable_count = stable_count + 1;

        try
            [y_k, t_k] = step(Tk, t);
            plot(t_k, squeeze(y_k), 'b');
            infos_unc{k} = stepinfo(Tk);
        catch
            disp(['Could not plot uncertain sample ', num2str(k), '.']);
        end
    else
        disp(['Sample ', num2str(k), ' closed loop is unstable.']);
    end
end

title('Nominal and Perturbed Closed-Loop Step Responses');
xlabel('Time (s)');
ylabel('\theta (rad)');
legend('Nominal','Uncertain samples','Location','best');

fprintf('Stable uncertain closed-loop samples: %d out of %d\n', stable_count, N);

%% ----------------------------------------------------------
% 12) Control effort response
% u/r = K*S
U_over_R = minreal(K*S_nom);

figure;
step(U_over_R, t);
grid on;
title('Nominal Control Effort for Unit Step Reference');
xlabel('Time (s)');
ylabel('\delta_e');

%% ----------------------------------------------------------
% 13) Reference tracking vs measurement-noise transmission
% For measurement noise entering at the sensor, y/n = -T
figure;
bodemag(Tr_nom, -T_nom);
grid on;
legend('Reference to Output','Noise to Output','Location','best');
title('Reference Tracking vs Measurement Noise Transmission');

%% ----------------------------------------------------------
```

```
% 14) Robust stability check
% To avoid failures I used try catch
CL_unc = minreal(feedback(G_unc*K, 1));

try
    [stabmarg, destabunc, report_stab] = robuststab(CL_unc);
    disp('Robust stability report:');
    disp(report_stab);
catch ME
    disp('robuststab could not be completed on this system/version.');
    disp(ME.message);
end

%% ------------------------------------------------------------
% 15) Robust performance check
try
    [perfmarg, worstunc, report_perf] = robustperf(CL_unc);
    disp('Robust performance report:');
    disp(report_perf);
catch ME
    disp('robustperf could not be completed on this system/version.');
    disp(ME.message);
end

%% ------------------------------------------------------------
% 16) Summary printed in command window
fprintf('\n--------------------------------------------------------\n');
fprintf('PROJECT SUMMARY\n');
fprintf('--------------------------------------------------------\n');
fprintf('Selected paper: Robust H-infinity Control Applied on a Fixed Wing UAV\n');
fprintf('Channel used    : Elevator to pitch angle\n');
fprintf('Method          : H-infinity mixed-sensitivity (mixsyn)\n');
fprintf('Nominal stable : %d\n', isstable(Tr_nom));
fprintf('Gamma achieved : %.4f\n', gamma_opt);
fprintf('Stable uncertain samples: %d / %d\n', stable_count, N);

if exist('stabmarg','var')
    try
        fprintf('Robust stability lower bound: %.4f\n', stabmarg.LowerBound);
        fprintf('Robust stability upper bound: %.4f\n', stabmarg.UpperBound);
    catch
```

```
        end
    end

    fprintf('-----------------------------------------------------\
        n');
```

Listing 1: MATLAB code for robust H_∞ pitch control of the fixed-wing UAV

References

[1] C. Uyulan and M. T. Yavuz, "Robust H_∞ control applied on a fixed wing unmanned aerial vehicle," *Advances in Aircraft and Spacecraft Science*, vol. 6, no. 5, 2019.