MAE 3260 Final Group Work: Exploring a System of Interest

Report

Outline:

Option #1: one report, same grade for all
Page 1: Cover page **Pages 2-3:** Block Diagram **Pages 3-5**: State Space Model **Pages 5-9:** Matlab Analysis **Page 10:** References

Title: Aircraft Pitch Closed Loop Control Analysis

Topic of Interest: Aircraft Pitch Control

Abstract: The system we are studying is the pitch control of an aircraft. We decided on this system because we are all interested in aeronautics. We plan to study the closed-loop control of this system and attempt to develop a simplified block diagram for pitch control in ideal conditions. This block diagram should include the direct reference inputs from the pilots in addition to exterior disturbances such as drag and sensor noise. Ultimately, the output of the system is theta, the pitch angle of the plane.

Students/Roles:

Student	**Task/Role**	**Portfolio**
Name	1-3 sentence summary of student's work and key take-aways	Link to results in student's individual portfolio
David Kaufman	Designed Transfer Function of Block Diagram of System	
Christine Guo	Design State Space for Plant in Block Diagram	
Zach Snarr	Analyze Open Loop & Closed Loop Dynamics (plant dynamics)	
Alexa Rosano	Contextualized physical components of the plane and drew block diagram	

List of MAE 3260 concepts or skills used in this group work:

- State Space Modeling
- Block Diagrams
- Transfer Functions
- Proportional Control with Disturbances

Block Diagram of Pitch Control System

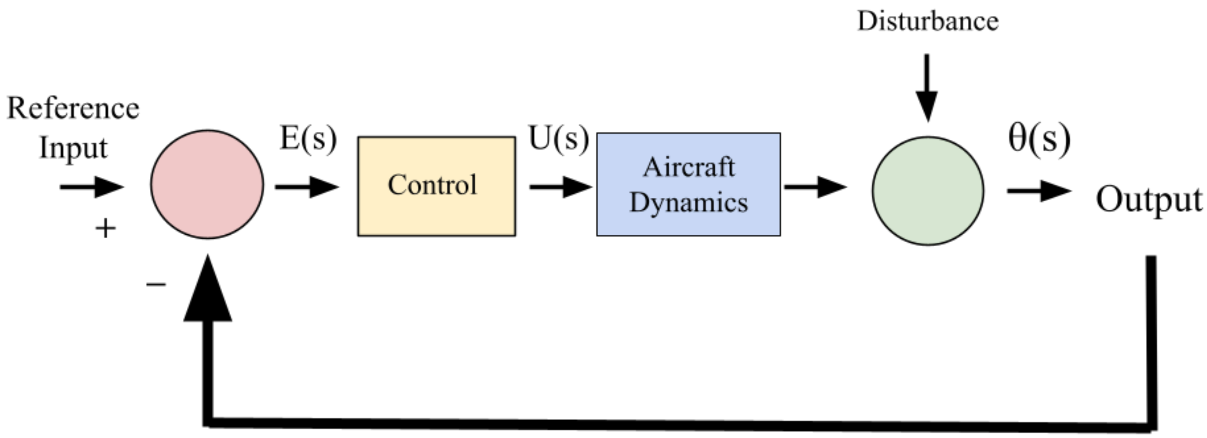

Figure 1: Block diagram of a closed-loop system modeling the pitch angle of an aircraft

The system we are analyzing is the pitch angle of an aircraft, and the control system (we chose a proportional control) that monitors that angle. Given a reference angle that the user wishes the aircraft to be in, and adding in a disturbance (in real life, this could be some wind, weather anomalies, drag forces, etc.), we can model a simplified scenario through the block diagram above.

Control Block:
Takes as an input the error E(s) that results from the disturbance that affects the pitch angle coming out of the aircraft dynamics block. This error is calculated from the difference between our reference input and the pitch angle, and is then fed into our proportional control system to give us the output U(s).

Aircraft Dynamics Block:
Contains a series of equations that define the pitch angle of the aircraft, where the input U(s) is defined as the elevator angle of the plane, and the output is the resulting pitch angle of the plane, which is then affected by the disturbances. See the section on aircraft dynamics state space below for a detailed analysis of the equations and their state space model.

General Transfer Functions for Block Diagram:
Controller:

$U(s) \; = \; KE(s)$

$E(s) \; = \; R(s) \; - \; \Theta(s)$

$U(s) \; = K_P E(s) \; = K_P[R(s) \; - \; \Theta(S)]$

Plane Output (before disturbance):

$\Theta_p(s) = P(s)U(s) = P(s)K_p[R(s) - \Theta(s)]$

Adding Disturbance:
$\Theta(s) = \Delta(s) + \Theta_P(s) = \Delta(s) + P(s)K_P[R(s) - \Theta(s)]$
where $\Delta(s)$ is the disturbance, and $\Theta_P(s)$ is the plant output

Solving:
$\Theta(s) + K_P P(s)\Theta(s) = K_P P(s)R(s) + \Delta(s)$
$\Theta(s) = \frac{K_P P(s)R(s)+\Delta(s)}{1+K_p P(s)} = \frac{K_P P(s)R(s)}{1+K_p P(s)} + \frac{\Delta(s)}{1+K_p P(s)}$

This can be split into two transfer functions:

1) Reference → Output (zero disturbance)
   - $T_R = \frac{\Theta(s)}{R(s)} = \frac{K_p P(s)}{1+K_p P(s)}$

2) Disturbance → Output (a zero reference angle, simulating steady level flight)
   - $T_D = \frac{\Theta(s)}{\Delta(s)} = \frac{1}{1+K_p P(s)}$

Altogether we get:
$\Theta(s) = T_R R(s) + T_D \Delta(s)$ where T_R and T_D are defined as above.

Plant (Aircraft Dynamics) State Space

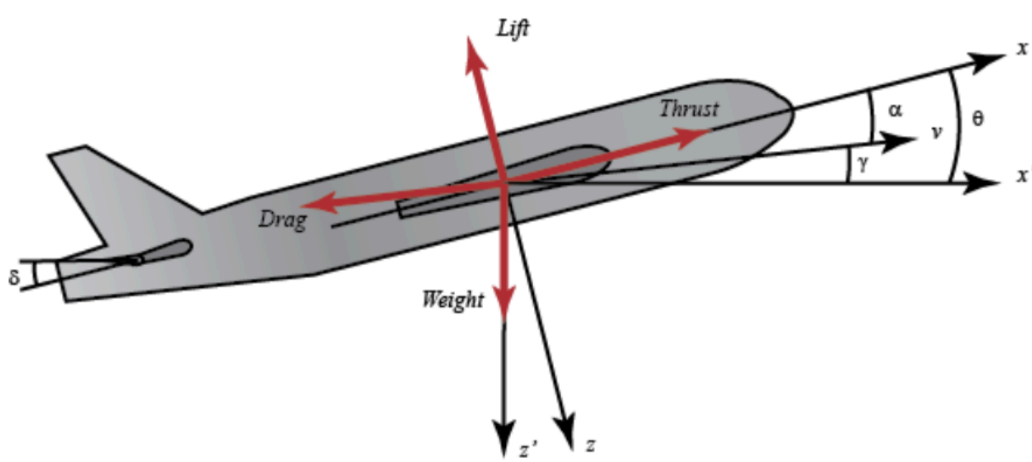

Figure 2: Diagram of forces and dimensions of an aircraft in flight
Credits: CTMS University of Michigan [1]

External Forces:

- Lift: Aerodynamic force perpendicular to the airflow, opposing weight
- Weight: Gravitational force acting downward
- Thrust: Forward propulsive force generated by the engine/propeller
- Drag: Aerodynamic force resisting forward motion

Dynamic Model:

The following equations [1] and variables [2] obtained from the University of Michigan's system modeling of aircraft pitch simplify the flight dynamics of angle of attack, pitch rate, and pitch angle for a plane configuration referenced in the figure above.

$$\dot{\alpha} = \mu\Omega\sigma[-(C_L + C_D)\alpha + \frac{1}{(\mu - C_L)}q - (C_W \sin\gamma)\theta + C_L]$$

$$\dot{q} = \frac{\mu\Omega}{2i_{yy}}[[C_M - \eta(C_L + C_D)]\alpha + [C_M + \sigma C_M(1 - \mu C_L)]q + (\eta C_W \sin\gamma)\delta]$$

$$\dot{\theta} = \Omega q$$

$$\mu = \frac{\rho S\bar{c}}{4m} \quad \Omega = \frac{2U}{\bar{c}} \quad \sigma = \frac{1}{1+\mu C_L} \quad \eta = \mu\sigma C_M \quad \delta = elevator\ pitch$$

The explanation of each coefficient and variable is listed below.

State Variables:

- α (angle of attack): Angle between the wing chord and the oncoming airflow.
- q (pitch rate): Rotational rate about the aircraft's lateral axis.
- θ (pitch angle): Aircraft's nose orientation relative to the horizontal.

Control Input (In the block diagram, this would be U(s)):

- δ (elevator deflection): Control surface angle commanding nose-up or nose-down motion.

Aerodynamic Coefficients:

- C_L (lift coefficient): Dimensionless measure of lift generated by the wing.
- C_D (drag coefficient): Dimensionless measure of aerodynamic drag.
- C_M (pitching-moment coefficient): Dimensionless measure of aerodynamic pitching torque.

- C_W (weight coefficient): Nondimensional representation of gravitational force.

Geometric and Mass-Related Parameters:

- μ (mass parameter): Nondimensional mass term derived from aircraft mass, wing area, and chord.
- η (eta, tail efficiency parameter): Aerodynamic coupling factor representing downwash/tail effectiveness.
- σ (sigma, moment scaling parameter): Geometric factor influencing pitching-moment aerodynamic behavior.
- i_{yy} (pitch moment of inertia): Aircraft's resistance to changes in pitch rate.
- $\bar{c}$ (mean aerodynamic chord): A calculated chord length for a tapered or swept wing that mimics a rectangular wing with the same aerodynamic effects as the original.

Flight-Condition Parameters:

- Ω (Omega, rotational scaling): Scaling factor relating nondimensional pitch-rate terms to dimensional dynamics.
- γ (flight-path angle): Angle between the velocity vector and the horizontal plane

Using these equations and variables, we can create a state space model:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -\mu\Omega\sigma(C_L + C_D) & \dfrac{\mu\Omega\sigma}{\mu - C_L} & -\mu\Omega\sigma C_W \sin\gamma \\ \dfrac{\mu\Omega}{2i_{yy}}(C_M - \eta(C_L + C_D)) & \dfrac{\mu\Omega}{2i_{yy}}(C_M + \sigma C_M(1 - \mu C_L)) & 0 \\ 0 & \Omega & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} \mu\Omega\sigma C_L \\ \dfrac{\mu\Omega}{2i_{yy}}(\eta C_W \sin\gamma) \\ 0 \end{bmatrix} [\delta]$$

$$\mathrm{y} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix}$$

Where the input is δ, and the output is what we are analyzing: the pitch angle of the aircraft, θ.

MatLab Script

Created with the help of ChatGPT [3] and using arbitrary numbers for aircraft parameters as an example.

Script analyzing the Open-Loop behavior of the state space model:

```
%% =====================================================================
%   Aircraft Pitch Dynamics - General Open-Loop Analysis (ASCII version)
%
```

```
%   State vector:
%       x = [ alpha ; q ; theta ]
%
%   Input:
%       u = delta_e       (elevator deflection)
%
%   Output:
%       y = theta
% ========================================================================
clear; clc; close all;
%% =======================================================================
% 1. USER INPUT: Physical Aircraft Parameters
% ========================================================================
params.m    = 9000;       % mass [kg]
params.Iyy = 1.0e6;       % pitch moment of inertia [kg*m^2]
params.U0   = 200;        % trim airspeed [m/s]
% Aerodynamic force derivatives (body Z-force)
params.Z_alpha = -5.0e4;  % dZ/d(alpha) [N/rad]
params.Z_q     = -1.0e5;  % dZ/d(q) [N/(rad/s)]
params.Z_de    =  2.0e4;  % dZ/d(delta_e) [N/rad]
% Aerodynamic pitching-moment derivatives
params.M_alpha = -2.0e6;  % dM/d(alpha) [N*m/rad]
params.M_q     = -8.0e6;  % dM/d(q) [N*m/(rad/s)]
params.M_de    = -3.0e6;  % dM/d(delta_e) [N*m/rad]
%% =======================================================================
% 2. Build State-Space Matrices
% ========================================================================
[A,B,C,D] = build_pitch_state_space(params);
disp('A matrix:'); A
disp('B matrix:'); B
disp('C matrix:'); C
disp('D matrix:'); D
sys_ss = ss(A,B,C,D);
%% =======================================================================
% 3. Step Response
% ========================================================================
figure;
step(sys_ss);
grid on;
title('Open-Loop Step Response: delta_e -> theta');
%% =======================================================================
% 4. Impulse Response
% ========================================================================
figure;
impulse(sys_ss);
grid on;
title('Open-Loop Impulse Response: delta_e -> theta');
%% =======================================================================
% 5. Transfer Function and Poles/Zeros
% ========================================================================
sys_tf = tf(sys_ss);
disp('Transfer Function G(s) = theta(s) / delta_e(s):');
sys_tf
figure;
pzmap(sys_ss);
grid on;
title('Open-Loop Pole-Zero Map');
%% =======================================================================
% 6. Frequency Response
% ========================================================================
figure;
bode(sys_ss);
grid on;
title('Open Loop Bode Plot');
%% =======================================================================
% Local Function
% ========================================================================
```

```
function [A,B,C,D] = build_pitch_state_space(p)
% Build state-space model using small-disturbance longitudinal equations.
m   = p.m;
Iyy = p.Iyy;
U0  = p.U0;
Z_alpha = p.Z_alpha;
Z_q     = p.Z_q;
Z_de    = p.Z_de;
M_alpha = p.M_alpha;
M_q     = p.M_q;
M_de    = p.M_de;
% Coefficients (small-disturbance longitudinal model)
a11 = Z_alpha / (m * U0);
a12 = (Z_q + m*U0) / (m * U0);
a21 = M_alpha / Iyy;
a22 = M_q     / Iyy;
b1 = Z_de / (m * U0);
b2 = M_de / Iyy;
% State-space matrices:
A = [ a11  a12  0 ;
     a21  a22  0 ;
     0     1   0 ];
B = [ b1 ;
     b2 ;
     0 ];
C = [0 0 1];     % Output = theta
D = 0;
end
```

Script Analyzing the Closed Loop System with P-Control & Disturbance:

```
%% ===========================================================================
%   Aircraft Pitch Dynamics - P-Controlled Closed Loop System
%
%   Output: theta (pitch angle)
%   Input:  delta_e (elevator deflection)
%
%   Controller: u = Kp * (theta_ref - theta)
%
%   This script:
%       1. Loads the given plant (A,B,C,D)
%       2. Computes open-loop TF P(s)
%       3. Forms closed-loop TF T(s) = feedback(Kp*P(s),1)
%       4. Builds closed-loop state-space (A_cl)
%       5. Simulates step tracking and disturbance rejection
% ===========================================================================
clear; clc; close all;
%% ===========================================================================
% 1. Plant Model (CTMS Aircraft Pitch Linearization)
% ===========================================================================
A = [-0.313   56.7     0;
    -0.0139 -0.426    0;
     0       56.7     0];
B = [0.232;
    0.0203;
    0];
C = [0 0 1];   % theta output
D = 0;
plant_ss = ss(A,B,C,D);
disp('Open-loop State-Space Model:')
A, B, C, D
%% ===========================================================================
% 2. Open-Loop Transfer Function P(s)
% ===========================================================================
P = tf(plant_ss);
disp('Open-loop Transfer Function P(s) = theta(s) / delta_e(s):')
P
```

```
%% ==========================================================================
% 3. Proportional Control Gain
% ==========================================================================
Kp = 1.0;      % <<< adjust this gain as needed
fprintf('\nUsing proportional gain Kp = %.3f\n', Kp);
%% ==========================================================================
% 4. Closed-Loop Transfer Function (theta_ref -> theta)
%
%   T(s) = feedback(Kp * P(s), 1)
% ==========================================================================
T = feedback(Kp * P, 1);
disp('Closed-loop Transfer Function T(s) = theta(s) / theta_ref(s):')
T
%% ==========================================================================
% 5. Closed-loop State-Space A_cl = A - B*Kp*C
% ==========================================================================
A_cl = A - B*Kp*C;
B_ref = B*Kp;
sys_cl = ss(A_cl, B_ref, C, 0);
disp('Closed-loop eigenvalues:')
eig(A_cl)
%% ==========================================================================
% 6. Step Response Tracking theta_ref
% ==========================================================================
t = 0:0.01:10;
theta_ref = ones(size(t));   % 1 radian step
[y_cl, t_cl] = lsim(sys_cl, theta_ref, t);
figure;
plot(t_cl, y_cl, 'LineWidth', 2); hold on;
yline(1, '--r', 'theta\_ref');
grid on;
xlabel('Time (s)');
ylabel('theta (rad)');
title(sprintf('Closed-Loop Pitch Tracking (P-Control, Kp = %.2f)', Kp));
legend('theta(t)', 'theta\_ref', 'Location', 'Best');
%% ==========================================================================
% 7. Optional: Disturbance Rejection Test
%
% Disturbance enters alpha dynamics: xdot = A_cl*x + Bref*theta_ref + E*d
% ==========================================================================
E = [1; 0; 0];    % simple disturbance affecting alpha dot
B_dist = E;
sys_cl_dist = ss(A_cl, [B_ref B_dist], C, [0 0]);
% Inputs: [theta_ref ; disturbance]
theta_ref = ones(size(t));
disturbance = 0.2*(t > 5);   % disturbance at t = 5s
inputs = [theta_ref' disturbance'];
[y_dist, t_dist] = lsim(sys_cl_dist, inputs, t);
figure;
plot(t_dist, y_dist, 'LineWidth', 2); hold on;
yline(1,'--r','theta\_ref');
xline(5,':k','Disturbance On');
grid on;
xlabel('Time (s)');
ylabel('theta (rad)');
title(sprintf('Disturbance Rejection (P-Control, Kp = %.2f)', Kp));
legend('theta(t)', 'theta\_ref');
```

Plots of Results:

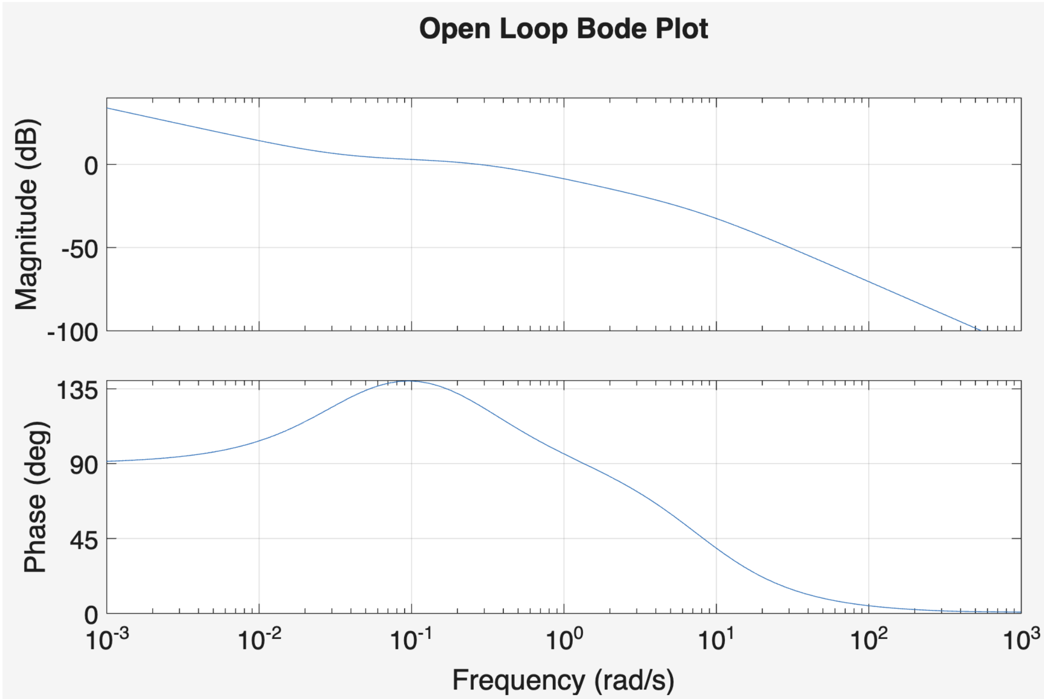

Figure 3: Bode plots of an example of our open-loop system depicting the behavior of our state space model. No control system has been applied in this scenario.

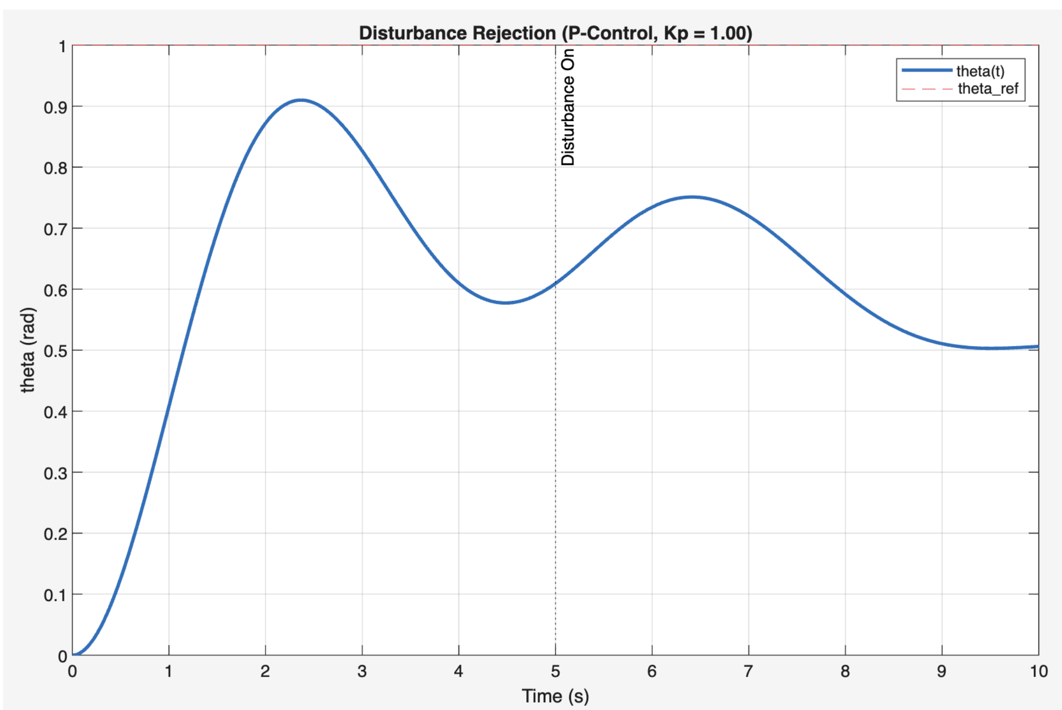

Figure 4: System response for our closed-loop system including disturbance. Theta in this case is our pitch angle of the plane.

References

[1] University of Michigan, Control tutorials for MATLAB and Simulink - aircraft pitch: System modeling, https://ctms.engin.umich.edu/CTMS/index.php?example=AircraftPitch&section=SystemModeling. 2010. (accessed Dec. 5, 2025).

[2] University of Michigan, Control tutorials for MATLAB and Simulink - aircraft pitch variables. 2010. https://ctms.engin.umich.edu/CTMS/index.php?aux=Extras_AircraftPitchVariables. (accessed Dec. 5, 2025).

[3] OpenAI, Chatgpt, https://chatgpt.com. 2022. (accessed Dec. 5, 2025).