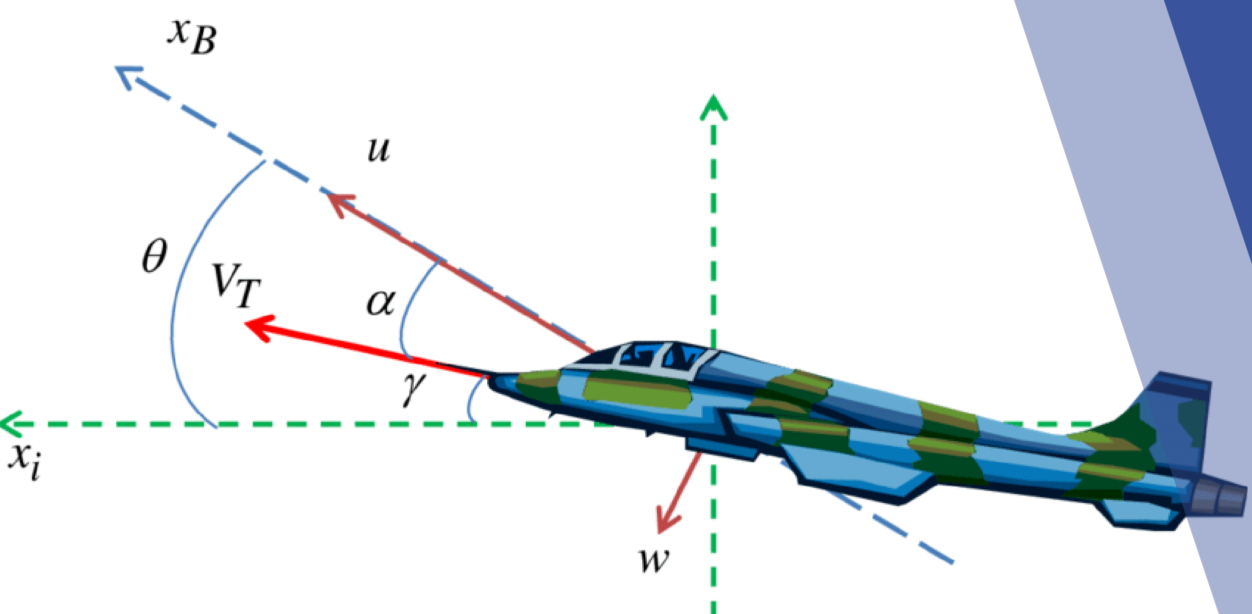
2025

AIRCRAFT LONGITUDINAL DYNAMICS SIMULATION

FLIGHT DYNAMICS AND CONTROL



PREPARED BY

- **MOHAMED HARHASH**
- **BARBARA KARAM**

SUPERVISED BY

DR. AHMED ABOULFTOUH

1. Introduction

This report details the simulation of an aircraft's longitudinal motion using a simplified six-state dynamic model. The code utilizes MATLAB's ODE solver ode45 to integrate the equations of motion (EOM) over time. The simulation captures key state variables such as forward and vertical velocities, pitch angle and rate, position, and angle of attack. This model is beneficial for initial stability and control analysis and provides a framework for further enhancements like control system integration or sensor simulation.

2. Initial Conditions

This section defines the initial physical parameters of the aircraft at time zero. The aircraft's speed, pitch angle, and initial position are specified, which are then decomposed into components.

- Code Snippet:

```
%% ------------------- Initial Conditions -------------------
V = 30;                         % Velocity of the aircraft (m/s)
theta0 = 2.1471;                % Initial pitch angle (deg.)
u0 = V * cosd(theta0);         % Initial forward velocity (m/s)
w0 = V * sind(theta0);         % Initial vertical velocity (m/s)
theta0_rad = deg2rad(theta0);  % Initial pitch angle (rad)
q0 = 0;                         % Initial pitch rate (rad/s)
x0 = 0;                         % Initial x-position (m)
z0 = 0;                         % Initial z-position (m)
```

3. Time Span of Simulation

The total time for simulation is set to 500 seconds, providing sufficient duration to analyze the aircraft's response.

- Code Snippet:

```
%% -------------------- Time Span --------------------
tspan = [0 500];                 % Simulate for 500 seconds
```

4. Numerical Integration Using ODE45

MATLAB's ode45 function is employed to solve the nonlinear system of equations defined in the EOM function.

- Code Snippet:

```
%% -------------------- ODE Solver --------------------
[t, states] = ode45(@EOM, tspan, initial_conditions);
```

5. State Extraction

Post-integration, the code extracts the state variables (u, w, theta, q, x, z) and computes the angle of attack alpha.

- Code Snippet:

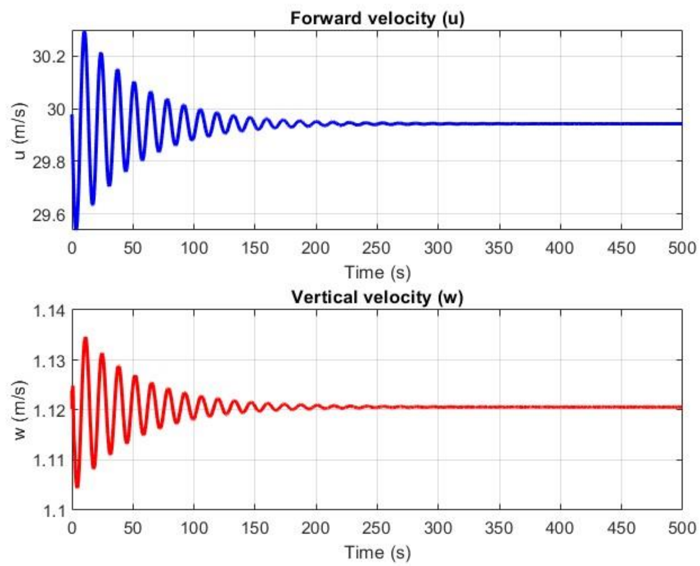
```
%% -------------------- Extract States --------------------
u     = states(:,1);
w     = states(:,2);
theta = states(:,3);
q     = states(:,4);
x     = states(:,5);
z     = states(:,6);
alpha = atan2(w, u);
```

6. Visualization of Simulation Results

6.1 Velocity Plots

Plots are generated for both the forward and vertical velocities.

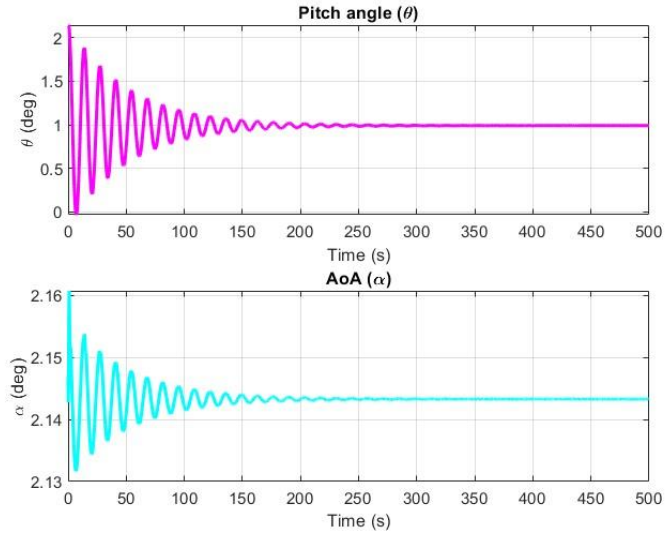
- Forward velocity (u) vs Time
- Vertical velocity (w) vs Time



6.2 Angular Response

Pitch angle (theta) and angle of attack (alpha) are plotted over time.

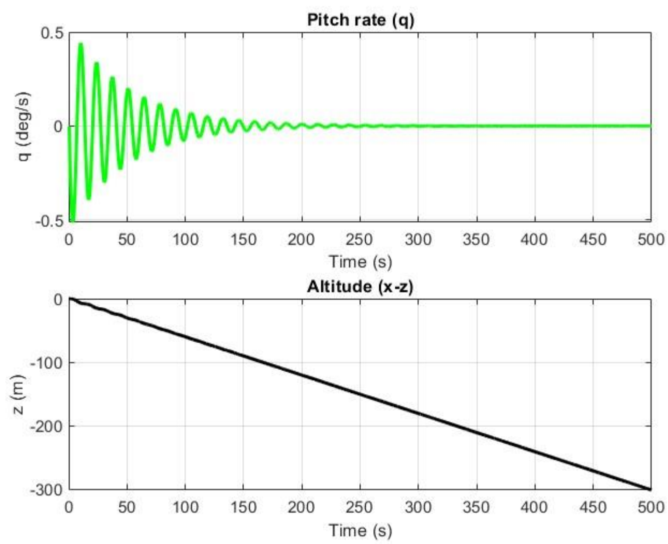
- Pitch angle (degrees) vs Time
- Angle of Attack (degrees) vs Time



6.3 Pitch Rate and Altitude

Two important performance parameters: pitch rate q and altitude change (represented by -z), are visualized.

- Pitch rate (degrees/sec) vs Time
- Altitude vs Time



7. Equations of Motion (EOM)

7.1 Assumptions

To formulate the equations governing the longitudinal motion of the aircraft, the following assumptions are made:

1. **2D Longitudinal Motion Only**: The analysis is confined to the x–z (vertical) plane. Lateral-directional dynamics such as roll, yaw, and sideslip are neglected.

2. **Rigid Body**: The aircraft is modeled as a rigid body with constant mass and moment of inertia. Structural flexibility is not considered.
3. **Flat, Non-Rotating Earth**: The Earth is assumed flat and non-rotating, so Coriolis and centrifugal forces are neglected.
4. **Symmetric Flight**: The aircraft motion is symmetric about the x–z plane, and there are no lateral or asymmetric effects.
5. **No Wind or Turbulence**: The atmospheric environment is assumed to be still, with no wind gusts or turbulence acting on the aircraft.
6. **Constant Atmospheric Properties**: Air density ρ and gravity g are assumed constant, representing standard sea-level conditions.

7.2 Dynamic Model

The longitudinal dynamics of the aircraft are modeled using a six-state nonlinear system. The state vector is:

$$\mathbf{y} = \begin{bmatrix} u & w & \theta & q & x & z \end{bmatrix}^T$$

The equations of motion are derived from Newton's second law in the body frame:

$$\begin{aligned}
\dot{u} &= \frac{F_x}{m} - qw \\
\dot{w} &= \frac{F_z}{m} + qu \\
\dot{\theta} &= q \\
\dot{q} &= \frac{M_y}{I_{yy}} \\
\dot{x} &= u\cos\theta + w\sin\theta \\
\dot{z} &= -u\sin\theta + w\cos\theta
\end{aligned}$$

7.3 Force and Moment Models

The angle of attack α and total air speed V are computed as:

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right), \quad V = \sqrt{u^2 + w^2}$$

The aerodynamic force and moment coefficients are expressed as linear functions of α and elevator deflection δe:

$$
\begin{aligned}
C_L &= C_{L_0} + C_{L_\alpha}\alpha + C_{L_{\delta_e}}\delta_e \\
C_D &= C_{D_0} + C_{D_\alpha}\alpha \\
C_M &= C_{M_0} + C_{M_\alpha}\alpha + C_{M_{\delta_e}}\delta_e
\end{aligned}
$$

These coefficients are then used to calculate lift L, drag D, and pitching moment My:

$$
\begin{aligned}
L &= \frac{1}{2}\rho V^2 S C_L \\
D &= \frac{1}{2}\rho V^2 S C_D \\
M_y &= \frac{1}{2}\rho V^2 S c C_M
\end{aligned}
$$

Thrust T is applied in the body x-direction and modeled as:

$$T = T_{\max} \cdot \delta_t$$

The aerodynamic forces in the body axes are:

$$
\begin{aligned}
F_x &= -D\cos\alpha + L\sin\alpha + T \\
F_z &= -L\cos\alpha - D\sin\alpha
\end{aligned}
$$

These are used to calculate accelerations in the body frame, accounting for gravity resolved along the aircraft's orientation:

$$
\begin{aligned}
A_{x_e} &= \frac{F_x}{m} - g\sin\theta \\
A_{z_e} &= \frac{F_z}{m} + g\cos\theta
\end{aligned}
$$

The resulting accelerations and rotational dynamics are integrated over time to simulate the aircraft’s motion using the numerical solver.

- Code Snippet:

```
%% -------------------- Equations of Motion --------------------
function dydt = EOM(t, y)
    % Constants
    m = 13.5; Iyy = 1.135; g = 9.81;
    rho = 1.225; S = 0.55; c = 0.19;
    T_max = 2 * g; delta_t = 0.5;

    % Aerodynamic coefficients
    CL0 = 0.28; CL_alpha = 3.45; CL_delta_e = -0.36;
    CD0 = 0.03; CD_alpha = 0.3;
    CM0 = -0.024; CM_alpha = -0.38; CM_delta_e = -0.5;

    % State extraction
    u = y(1); w = y(2); theta = y(3); q = y(4); x = y(5); z = y(6);
    V = sqrt(u^2 + w^2);
    alpha = atan2(w, u);
    delta_e = -4.3791 * (pi/180);

    % Aerodynamic forces and moments
    CL = CL0 + CL_alpha * alpha + CL_delta_e * delta_e;
    CD = CD0 + CD_alpha * alpha;
    CM = CM0 + CM_alpha * alpha + CM_delta_e * delta_e;

    L = 0.5 * rho * V^2 * S * CL;
    D = 0.5 * rho * V^2 * S * CD;
    My = 0.5 * rho * V^2 * S * c * CM;
    T = T_max * delta_t;

    % Forces in body axes
    Fx = -D * cos(alpha) + L * sin(alpha) + T;
    Fz = -L * cos(alpha) - D * sin(alpha);

    % Accelerations
    A_xe = Fx/m - g * sin(theta);
    A_ze = Fz/m + g * cos(theta);
    % State derivatives
    du     = A_xe - q * w;
    dw     = A_ze + q * u;
    dtheta = q;
    dq     = My / Iyy;
    dx     = u * cos(theta) + w * sin(theta);
    dz     = -u * sin(theta) + w * cos(theta);

    dydt = [du dw dtheta dq dx dz]';
end
```

8. Conclusion

This simulation successfully demonstrates how the longitudinal dynamics of an aircraft can be modeled and analyzed using MATLAB. The plots provide insights into system stability and dynamic behavior over time. This framework is extendable for advanced modeling such as control systems, sensor integration, or trajectory optimization.

9. Source Code

The complete source code used for this simulation is available on GitHub for reference and further exploration. You can access it using the following link:
Aircraft Dynamics Simulation - GitHub