TMME50 Assignment I

David Wiman
20000120-8495

November 24, 2023

Instructions 2023 for reporting the computer assignments

The computer assignments are reported in writing, and submitted *printed on paper*. The assignments are performed individually. The use of ChatGPT or any similar system is not allowed. It is permissible to discuss the assignments and to show parts of solutions in that context, but *copying of Matlab code or sections of reports is not allowed.* Further, it is not allowed to possess copies of other students reports or Matlab code, either electronically or on paper, or to supply this to another student; this also means that you hand in and pick up your assignments yourself, not with the help of a friend. The reports shall contain:

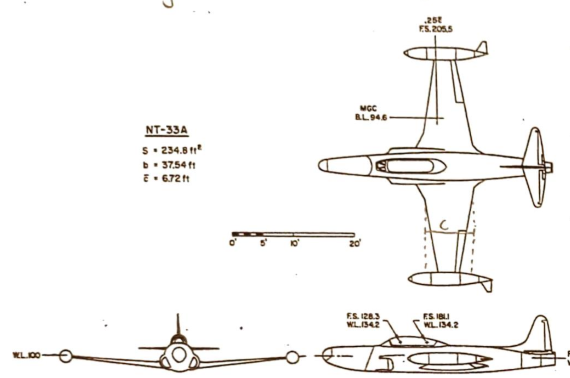
- *A copy of this page of instructions.*
- Name and complete (10 digits) civic registration number of the student (sometimes called p-number among exchange students).
- *Which aeroplane and which reference condition* that has been used. Specify the number of the column on the data sheet that has been used.
- Answers to all the questions appearing under the headings "Assignment I:a" etc. and all requested plots.
- A complete set of Matlab files for each computer assignment. Choose the most complete set, such as the one for part I:c in assignment I. In assignment II, also also include root loci and a graphic representation of your Simulink model for the final version of your model with all numerical values shown explicitly.
- The ODE system implemented in assignments I, III, IV and V must be given in the report in the order actually implemented and written in a *single* frame containing *all* the equations of the ODE and *nothing* else.

Further, note:

- With the exeption of flying qualities tables, illustrations from the lab-PM defining the problem and this page of instructions, no copying of text, figures, equations or code from another document is allowed (unless it is a document you have created yourself).
- It must be clear what data has been used in what way. Data is converted from American to SI units, and this should be done in a way that can be followed in detail either in the text of the report or in the Matlab files, so that misstakes can be found at a glance without making any calculations.
- Nothing written in Matlab syntax or pseudocode is allowed in the main text of the reports: your Matlab code is appended at the end of the report.
- The report must contain sufficient text and illustrations such that it is possible to understand without ever having seen the lab-PM.
- Use the simulation time given in the assignments. For a small number of datasets it is necessary to use a longer simulation time than 100 s in order to to see a full phugoid period, but the time should never be shorter than the time given and never longer than 400 s.

T-33 Shooting Star

David Wiman



h
V, u_0
m
I_x
I_y
I_z
I_{xz}
Q

surface, span, chord

$S = 234,8\ ft^2$, $b = 37,54\ ft$, $\bar{c} = 6,72\ ft$

	1	2	3	4	5	6	7	8
h ft	S/L	S/L	S/L	S/L	20000	20000	20000	40000
V(u_0) ft/s	228	270	447	782	311	570	778	629
m slug	367	426	426	426	426	426	426	426
I_x slug·ft²	12700	23801	23801	23801	23801	23801	23801	23801
I_y slug·ft²	20700	21101	21101	21101	21101	21101	21101	21101
I_z slug·ft²	32001	43802	43802	43802	43802	43802	43802	43802
I_{xz} slug·ft²	480	480	480	480	480	480	480	480
Q lb/ft²	61,7	86,7	247	726	61,3	206	383	117

	1	2	3	4	5	6	7	8
X_u s^{-1}	−0,0391	−0,00484	−0,0104	−0,0415	0,00477	−0,00735	−0,0511	−0,0355
X_α ft/s²	18,58	35,37	25,12	−16,50	20,43	22,29	7,67	24,59
Z_u s^{-1}	−0,248	−0,153	−0,128	−0,162	−0,114	−0,107	−0,0703	−0,0766
Z_α ft/s²	−213,41	−267,57	−773,31	−2807	−140,26	−712,5	−1400	−437,8
M_u $ft^{-1}s^{-1}$	0,000318	0,000603	0,000283	−0,00076	0,000114	−0,000183	−0,00151	−0,000183
M_α s^{-2}	−1,89	−1,81	−9,21	−33,7	−0,23	−9,95	−18,59	−5,42
$M_{\dot\alpha}$ s^{-1}	−0,35	−0,40	−0,63	0	−0,24	−0,31	−0,16	−0,06
M_q s^{-1}	−0,694	−0,806	−1,37	−2,80	−0,50	−0,981	−1,56	−0,535
X_{δ_e} ft/s²	0,516	1,47	0,62	−2,65	1,88	0,50	−0,432	0,996
Z_{δ_e} ft/s²	−13,4	−16,2	−44,4	−152	−11,3	−40,9	−82,4	−23,8
M_{δ_e} s^{-2}	−4,19	−5,83	−16,0	−52,7	−4,13	−14,2	−28,7	−8,28

	1	2	3	4	5	6	7	8
Y_β ft/s²	−28,4	−30,1	−81,0	−264	−21,6	−72,7	−144	−42,4
L_β s^{-2}	−5,49	−4,72	−8,02	−18,0	−4,06	−7,42	−9,89	−5,08
L_p s^{-1}	−2,03	−1,32	−2,15	−4,51	−0,82	−1,56	−2,23	−0,877
L_r s^{-1}	0,641	0,305	0,320	0,495	0,214	0,256	0,328	0,179
N_β s^{-2}	0,667	0,94	2,71	10,6	0,54	2,60	6,24	1,68
N_p s^{-1}	−0,116	−0,112	−0,0512	0,0118	−0,103	−0,0393	−0,0141	−0,0428
N_r s^{-1}	−0,207	−0,173	−0,291	−0,561	−0,104	−0,204	−0,318	−0,110
Y_{δ_r} ft/s²	0,0295	0,0301	0,0503	0,102	0,0185	0,0363	0,0571	0,0195
L_{δ_r} s^{-2}	−0,0125	0,443	1,57	5,89	0,287	1,39	3,20	0,808
N_{δ_r} s^{-2}	−1,24	−1,25	−3,50	−12,6	−0,883	−3,21	−6,99	−1,92
L_{δ_a} s^{-2}	6,01	4,53	12,6	47,0	3,14	11,7	24,0	7,13
N_{δ_a} s^{-1}	0,0286	0,134	0,165	0,260	0,164	0,121	0,195	0,118

Background

The motion of an aeroplane moving in the xz-plane is to be studied. We introduce one coordinate system xyz attached to the aeroplane and one coordinate system $x_f y_f z_f$ fixed to the ground and to be assumed to be inertial.

The longitudinal equations of motions for an aeroplane, together with kinematical relations for the orientation and position of the aeroplane can be written as

$$
\begin{aligned}
\dot{u} &= -qw - g\sin(\theta) + \frac{X}{m}, \\
\dot{w} &= qu + g\cos(\theta) + \frac{Z}{m}, \\
\dot{q} &= \frac{M}{I_{yy}}, \\
\dot{\theta} &= q, \\
\dot{x}_f &= u\cos(\theta) + w\sin(\theta), \\
\dot{z}_f &= u\sin(\theta) - w\cos(\theta).
\end{aligned} \tag{1}
$$

X, Z and M are forces and moments on the aeroplane, not including the force of gravity.

The data set for the tasks at hand were given in imperial units and must therefore be converted to SI units before use. Two of the values given were:

$$
\begin{gathered}
V = 228\ \text{ft/s} \cdot 0.3048 = 69.4944\ \text{m/s}, \\
H = 100\ \text{m}.
\end{gathered} \tag{2}
$$

I use data from a T-33 Shooting Star at sea level flight (column one on the data sheet).

I:a

Using $X = Z = M = 0$ and the initial conditions

$$
\begin{aligned}
u_i &= V, \\
w_i &= 0, \\
q_i &= 0, \\
\theta_i &= 0, \\
x_f^i &= 0, \\
z_f^i &= H,
\end{aligned} \tag{3}
$$

the system in (1) was simulated for 100 seconds. The plots in Figures 1 and 2 was captured.

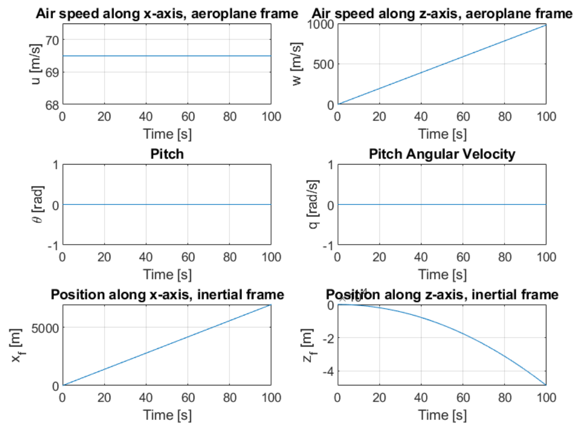


Figure 1: Phase variables.

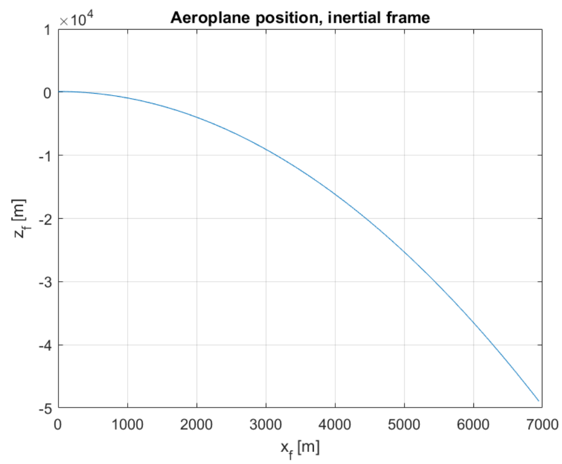


Figure 2: Aeroplane position in fixed frame.

We can see that the trajectory is a parabola that extends bellow the ground. The endpoint of the numerical solution is $x_f = 6949$ and $z_f = -48950$. This is in agreement with analytical solutions. These can be seen in (4).

$$
\begin{gathered}
x_f = \int_0^{100} V \; dt = 100 \cdot 69.4944 = 6949.44. \\
z_f = H - \int_0^{100} \left(\int w \; dt \right) dt = H - \int_0^{100} gt \; dt = 100 - 10000 \frac{9.81}{2} = -48950.
\end{gathered} \tag{4}
$$

I:b

Using the reference state

$$
\begin{aligned}
u_0 &= V, \\
w_0 &= 0, \\
q_0 &= 0, \\
\theta_0 &= 0, \\
x_f^0 &= 0, \\
z_f^0 &= H,
\end{aligned}
\tag{5}
$$

we can compute the aerodynamic forces in the reference state,

$$
\begin{aligned}
X_0 &= mg\sin(\theta_0) = 0, \\
Z_0 &= -mg\cos(\theta_0) = -mg, \\
\frac{M_0}{I_{yy}} &= 0.
\end{aligned}
\tag{6}
$$

With the forces in (6), another simulation was run for 100 seconds using the initial conditions in (3). The plots in Figures 3 and 4 was captured.

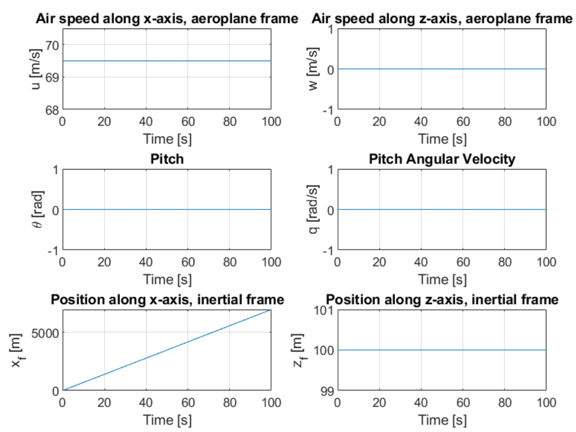


Figure 3: Phase variables.

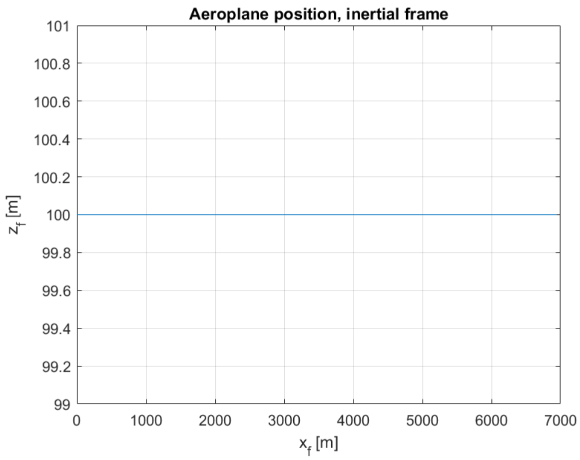


Figure 4: Aeroplane position in fixed frame.

We can clearly see in Figure 4 that the aeroplane flies in a straight line parallel to the ground.

I:c

Assume that the aerodynamic forces on the aeroplane can be written as

$$
\begin{aligned}
X_0 &= mg\sin(\theta_0) + mX_u(u-u_0) + mX_w(w-w_0),\\
Z_0 &= -mg\cos(\theta_0) + mZ_u(u-u_0) + mZ_w(w-w_0),\\
M_0 &= I_{yy}M_w(w-w_0) + I_{yy}M_q(q-q_0).
\end{aligned}
\tag{7}
$$

This is a linear model and may assume to be valid close to the reference state.

If we simulate the system with the forces modeled as in (7) and with the initial condition

$$
\begin{aligned}
u_i &= V,\\
w_i &= 0,\\
q_i &= 0,\\
\theta_i &= 0.1\text{rad},\\
x_f^i &= 0,\\
z_f^i &= H,
\end{aligned}
\tag{8}
$$

for 100 seconds, we get the plots in Figures 5 to 7.

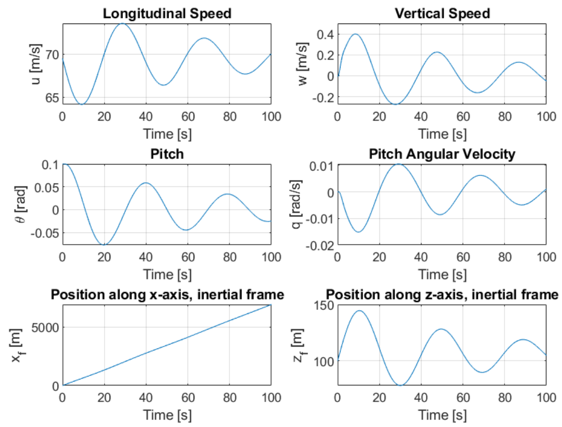


Figure 5: Phase variables.

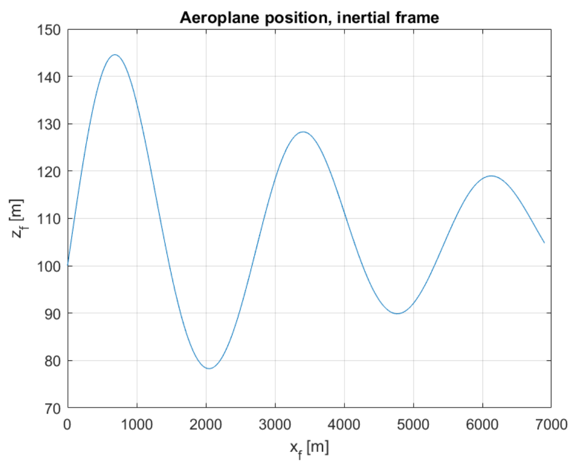


Figure 6: Aeroplane position in fixed frame.

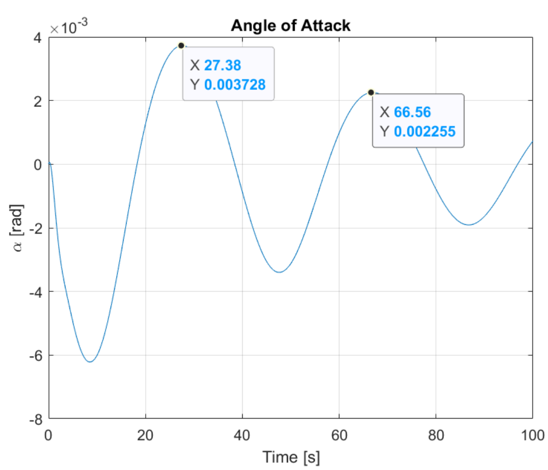


Figure 7: Angle of attack.

We can see that the aeroplane oscillates slightly in pitch in the beginning, but that this motion is dampened. The period time of the oscillation (the phugoid period) is computed by measuring the time difference between peaks in the angle of attack.

$$T_s = 66.56 - 27.38 = 39.18. \tag{9}$$

We also plot a close up view of the phase variables u and θ to be able to determine phase shift. This can be seen in Figure 8.

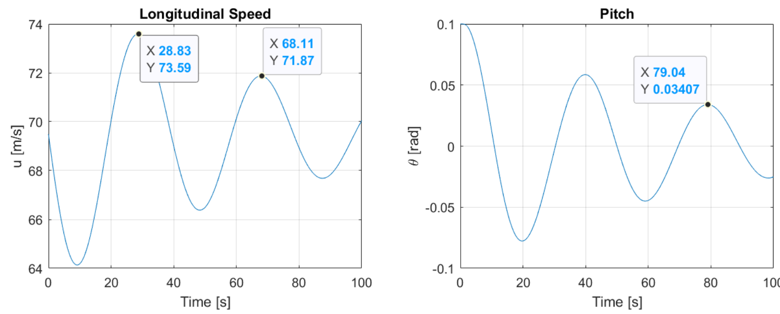


Figure 8: u and θ with data points.

The phase shift is computed by looking at the time difference between peaks in u and θ and then multiplying the result with 2π and dividing by the period time.

$$\varphi = 2\pi \frac{79.04 - 68.11}{68.11 - 28.83} = 1.748 \text{ rad} = 100.17° \tag{10}$$

If we include pictures in of the aeroplane in Figure 6 showing the angle of attack, we get Figure 9

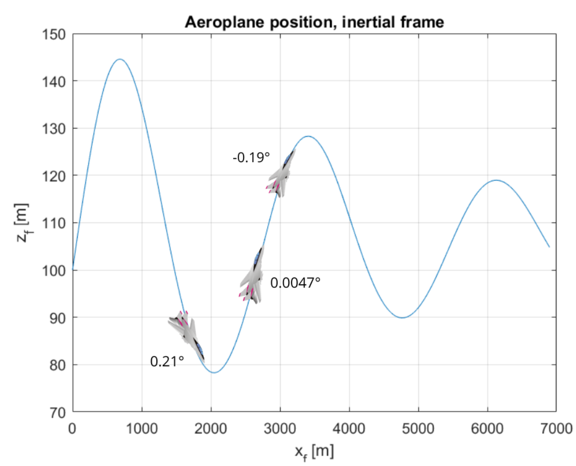


Figure 9: Aeroplane position with angle of attack shown.

From the plots, we can determine the AOA at certain x_f values. These are drawn in the graph. Drawing the aeroplanes accurately is extremely hard due to scaling but the relative size of the angles is shown as well as the sign.

The aeroplane starts with a positive pitch and immediately gains altitude. Since the aeroplane is statically stable, there is a restoring moment which returns the pitch to zero without elevator action. However, this motion has an overshoot and the pitch becomes negative, causing the aeroplane to begin descending. This is repeated with smaller and smaller overshoots until the aeroplane has returned to level flight.

Shortly after the initial pitch down, the AOA will be negative for a short while until the velocity vector has converged with the pitch vector and the AOA is returned to zero. This is because it takes time to change the velocity vector thanks to inertia. Then, when the aeroplane pitches up, the AOA will be positive until convergence. This repeats itself until level flight is achieved.

I:d

If we replace M_w with $|M_w|$ and redo the simulation with initial conditions (11) for 100 seconds, we get the plots in Figures 10 to 12.

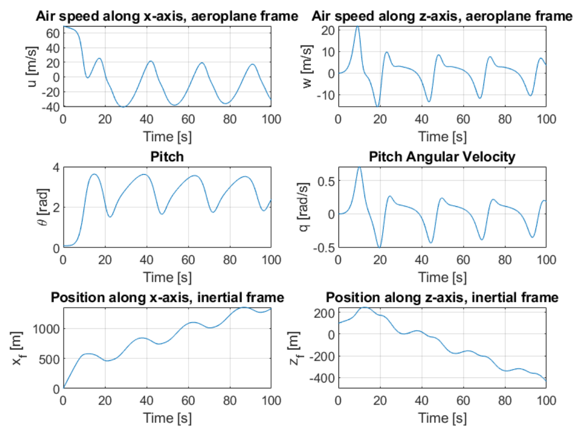


Figure 10: Phase variables.

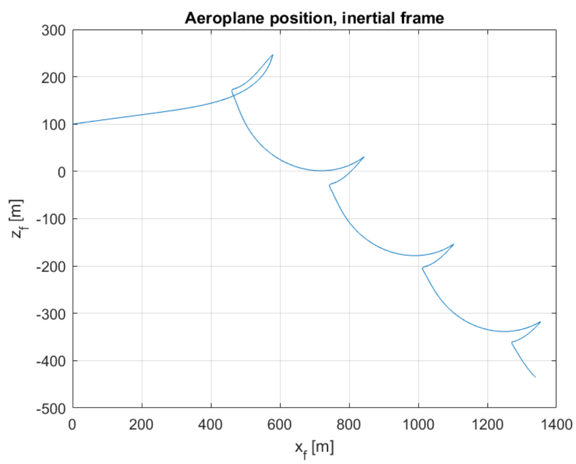


Figure 11: Aeroplane position in fixed frame.

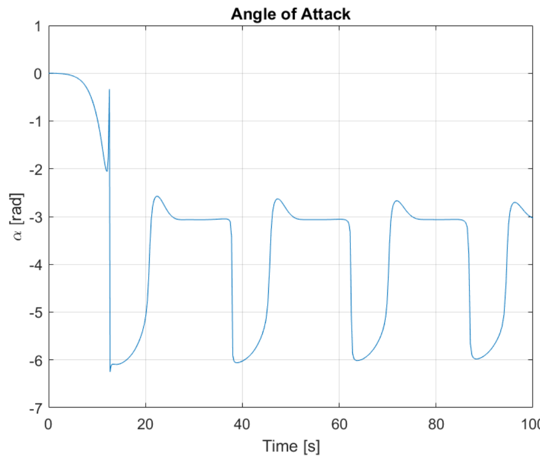


Figure 12: Angle of attack.

The angle of attack oscillates weirdly and the position of the aeroplane spirals downwards. Changing the sign of M_w ruins the flying capabilities of the aeroplane.

I:e

If we restore M_w to its original value but instead let $M_q = Z_w = 0$ and use the initial conditions

$$\begin{aligned} u_i &= V, \\ w_i &= 0, \\ q_i &= 0.1\text{rad/s}, \\ \theta_i &= 0, \\ x_f^i &= 0, \\ z_f^i &= H, \end{aligned} \tag{11}$$

and simulate the system for about one fifth of the phugoid oscillation period (8 seconds), we get the results in Figures 13 to 15.

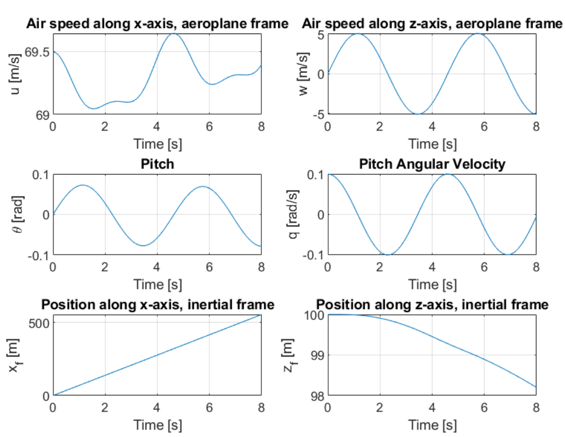


Figure 13: Phase variables.

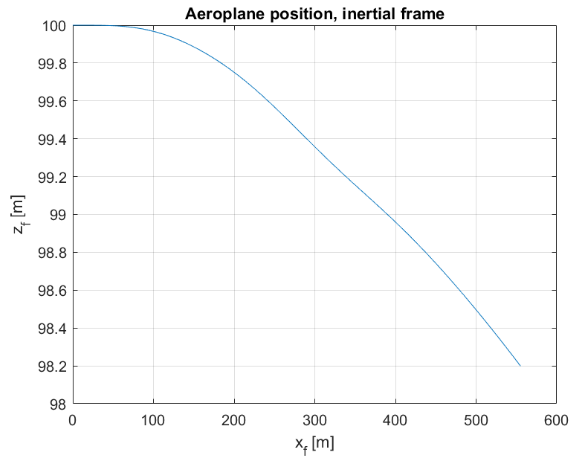


Figure 14: Aeroplane position in fixed frame.

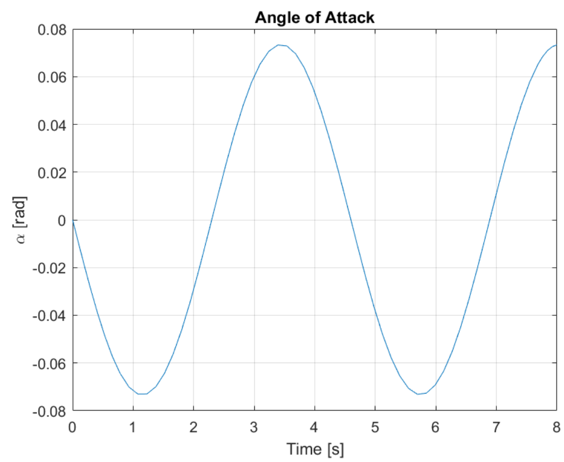


Figure 15: Angle of attack.

The short period motion differs from the phugoid motion since the phase difference between the pitch and the AOA is closer to 180° instead of 90°. This leads to that in I:c, the pitch starts by decreasing and so does the AOA for previously mentioned reasons. In this exercise, the pitch starts be increasing, but the AOA still decreases. There is a sign difference.

The short period motion is a high frequency, relatively large amplitude (~ 20 times larger than phugoid amplitude) nose oscillation. However, it does not severely effect the velocity direction.

The short period motion is also strongly dampened and quickly dies out.

1 Matlab code

1.1 Main file, I:c

```
% Set variables
run('T33_Shooting_Star_parameters.m');

% Reference state around which linearization happens
reference_state = [v, 0, 0, 0, 0, h];

% Force linearization parameters, [Xu, Xw, Zu, Zw, Mw, Mq]
force_lin_param = [Xu, Xw, Zu, Zw, Mw, Mq];

% Set ODE options
initial_theta = 0.1;
initial_conditions = [v, 0, 0, initial_theta, 0, h];
time_interval = [0, 100];

% Compute ODE
[time, state] = ode45(@(t, state) state_propagation_1_c(t, state, m, I_y,
    reference_state, force_lin_param), time_interval, initial_conditions);

% Extract all states for plotting
u = state(:,1);
w = state(:,2);
q = state(:,3);
theta = state(:,4);
x_f = state(:,5);
z_f = state(:,6);

% Plot
figure(1)
subplot(3,2,1)
plot(time, u);
title('Air speed along x-axis, aeroplane frame')
xlabel('Time [s]')
ylabel('u [m/s]')
grid

subplot(3,2,2)
plot(time, w);
title('Air speed along z-axis, aeroplane frame')
xlabel('Time [s]')
ylabel('w [m/s]')
grid

subplot(3,2,3)
plot(time, theta);
title('Pitch')
xlabel('Time [s]')
ylabel('\theta [rad]')
grid

subplot(3,2,4)
```

```
plot(time, q);
title('Pitch Angular Velocity')
xlabel('Time [s]')
ylabel('q [rad/s]')
grid

subplot(3,2,5)
plot(time, x_f);
title('Position along x-axis, inertial frame')
xlabel('Time [s]')
ylabel('x_f [m]')
grid

subplot(3,2,6)
plot(time, z_f);
title('Position along z-axis, inertial frame')
xlabel('Time [s]')
ylabel('z_f [m]')
grid

figure(2)
plot(x_f, z_f)
title('Aeroplane position, inertial frame')
xlabel('x_f [m]')
ylabel('z_f [m]')
grid

% Angle of Attack
speed_along_fixed_x = u.*cos(theta) + w.*sin(theta);
speed_along_fixed_z = u.*sin(theta) - w.*cos(theta); % Positive up
velocity_vector_angle = atan2(speed_along_fixed_z, speed_along_fixed_x);
alpha = velocity_vector_angle - theta; % Stange, should be other way around

figure(3)
plot(time, alpha)
title('Angle of Attack')
xlabel('Time [s]')
ylabel('\alpha [rad]')
grid
```

1.2 State propagation, I:c

```
% Function to propagate the state
function next_state = state_propagation_1_c(t, state, m, I_yy, reference_state
    , force_lin_param)
    % Rename state components for readability
    u = state(1);
    w = state(2);
    q = state(3);
    theta = state(4);
    x_f = state(5);
    z_f = state(6);

    % Extract reference state
    u0 = reference_state(1);
    w0 = reference_state(2);
    q0 = reference_state(3);
    theta_0 = reference_state(4);
    x_f_0 = reference_state(5);
    z_f_0 = reference_state(6);

    % Extract force linearization parameters
    Xu = force_lin_param(1);
    Xw = force_lin_param(2);
    Zu = force_lin_param(3);
    Zw = force_lin_param(4);
    Mw = force_lin_param(5);
    Mq = force_lin_param(6);

    % Constants
    g = 9.81;

    % Forces
    X = m*g*sin(theta_0) + m*Xu*(u - u0) + m*Xw*(w-w0);
    Z = -m*g*cos(theta_0) + m*Zu*(u - u0) + m*Zw*(w - w0);
    M = I_yy*Mw*(w - w0) + I_yy*Mq*(q - q0);

    % State dynamics
    next_state(1) = -q*w - g*sin(theta) + X/m;
    next_state(2) = q*u + g*cos(theta) + Z/m;
    next_state(3) = M/I_yy;
    next_state(4) = q;
    next_state(5) = u*cos(theta) + w*sin(theta);
    next_state(6) = u*sin(theta) - w*cos(theta);

    % Transpose to get row vector
    next_state = next_state';
end
```

1.3 Unit conversion

```
% Constants
g = 9.81;

% Unit conversions
feet_to_meter = 0.3048;
feet2_to_m2 = 0.09290;
lb_to_kg = 0.4536;
slug_to_kg = 14.59;
slug_feet2_to_kg_m2 = 1.356;
lb_to_N = 4.448;

% Parameters in SI units
h = 100;
v = 228*feet_to_meter;
m = 367*slug_to_kg;
I_x = 12700*slug_feet2_to_kg_m2;
I_y = 20700*slug_feet2_to_kg_m2;
I_z = 32001*slug_feet2_to_kg_m2;
I_xz = 480*slug_feet2_to_kg_m2;
Q = 61.7*lb_to_N/feet2_to_m2;

Xu = -0.0391;
Xalpha = 18.58*feet_to_meter;
Xw = Xalpha/v;
Zu = -0.248;
Zalpha = -213.41*feet_to_meter;
Zw = Zalpha/v;
Mu = 0.000318/feet_to_meter;
Malpha = -1.89;
Mw = Malpha/v;
Malpha_dot = -0.35;
Mw_dot = Malpha_dot/v;
Mq = -0.694;
Xdelta_e = 0.516*feet_to_meter;
Zdelta_e = -13.4*feet_to_meter;
Mdelta_e = -4.19;

Ybeta = -28.4*feet_to_meter;
Lbeta = -5.49;
Lp = -2.03;
Lr = 0.641;
Nbeta = 0.667;
Np = -0.116;
Nr = -0.207;
Ydelta_r = 0.0295*feet_to_meter;
Ldelta_r = -0.0125;
Ndelta_r = -1.24;
Ldelta_a = 6.01;
Ndelta_a = 0.0286;

S = 234.8*feet2_to_m2;
b = 37.54*feet_to_meter;
```

```
c = 6.72*feet_to_meter;
```