Creating MATLAB simulations to model aircraft dynamics

Joshua Lynch
Group 07
fk18538@bristol.ac.uk

Vedang Joshi
Group 07
vedang.joshi.2018@bristol.ac.uk

Vincent Peng
Group 07
ok18348@bristol.ac.uk

Elfa Ren
Group 07
uk18700@bristol.ac.uk

December 20, 2019

Abstract

The complexity in modelling aircraft pitch dynamics arises from the need to understand non-linear modelling. We use simple Euler method approximations, in two of the three models we present, and MATLAB's powerful ODE solver in one of the models. We analyse how pitch angles and linear velocities change in time in the body axes of the aircraft and talk about how unstable motion demonstrates "phugoid" and "short period" behaviour. We demonstrate our models on light aircraft types i.e. Cessna 172 and 182. We present the entire code used to simulate all models and unlike other studies in literature, which reduces the repeatability of results.

1 Introduction

Since the invention of aeroplanes, the aviation industry has always required the motion of the aircraft to be studied thoroughly to ensure passenger and crew safety, though flight simulations were initially only used to train pilots [12]. This has radically changed since the advent of NASA's 6 degrees of freedom (DOF) flight simulator [14]. The motivation for this study is to be able to predict simple forward flight and the pitch of the rigid body through ordinary differential equations (ODEs). We also establish the relationship between the linear velocities of the aircraft and the pitch angles on the motion of the aircraft. We will attempt to produce different models, each with more parameters to consider than the last, as a way to demonstrate the complexity of the calculations required for rigid body flight. Such work has already been undertaken considering constant wind velocities [7] or by modelling their motion for hovering aircraft [8] or by comparing their motion to those of flying insects [17]. Yet, the models published in literature are complex non-linear models, so we try to simplify these models as much as possible. We give the complete MATLAB code for each simulation to ease repeatability and attempt to give an accurate intuition as to the dynamics of an aircraft.

2 Classical mechanics: Newton's laws of motion

2.1 Euler approximation to the solution

The aircraft we take into consideration has its x-axis going through the nose of the aircraft and points forward, the z-axis is perpendicular to the x-axis and points downwards and the y-axis is perpendicular to the x-axis and points out of the right wing as shown in Figure 1. The inherent assumptions in setting this framework are that the earth is an inertial frame of reference and that the aircraft is a rigid body. We further assume that the aircraft flies in equilibrium condition for the duration of the flight and thus the linearisation of equations will be about these working flight conditions.

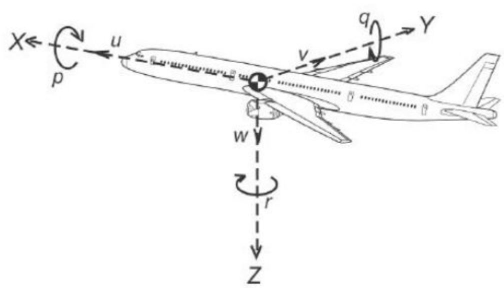

Figure 1: The body axes determined for the aircraft giving the 3 translational degrees of freedom [6]

If the aircraft is a rigid body, then using the relative velocity equation for rigid bodies,

$$\vec{v}_Q = \vec{v}_P + \vec{\omega} \times \vec{v}_{PQ} \tag{1}$$

for two particles P, Q. Taking the linear components of the velocity (u, v, w) and the rotational components for $\vec{\omega}$ given as (p, q, r),we can write the force equations for the rigid body. From Newton's second law and Eq.1 [13],

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} + \begin{pmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{pmatrix} \tag{2}$$

$$\vec{a} = \begin{pmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{pmatrix} \tag{3}$$

We only consider the pitch angle θ to be the only rotational angle in the system; the roll and yaw angles are taken to be zero in this first model of aircraft motion. The pitch rotation matrix is defined [15] as follows,

$$\vec{\theta} = \begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix} \tag{4}$$

The components of the gravitational acceleration in the body fixed system are as follows,

$$\begin{pmatrix} g_x \\ g_y \\ g_z \end{pmatrix} = \begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ g_0 \end{pmatrix} = \begin{pmatrix} -g_0 sin\theta \\ 0 \\ g_0 cos\theta \end{pmatrix} \tag{5}$$

Note that the only component of g i.e g_0 is in the z-direction. The generalised force equations are thus,

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + mg_0 \begin{pmatrix} -sin\theta \\ 0 \\ cos\theta \end{pmatrix} = m \begin{pmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{pmatrix} \tag{6}$$

As we are only concerned about the forward motion in the first model, we only analyse the first of the three equations of motion

$$X - mg_0 sin\theta = m(\dot{u} + qw - rv) \tag{7}$$

Solve approximations for,

$$X_0 + \Delta X - mg_0 sin(\theta_0 + \theta) = m(\dot{u} + qw - rv) \tag{8}$$

Using small angle approximations,

$$sin(\theta_0 + \theta) \approx sin\theta_0 + \theta cos(\theta) \tag{9}$$

Neglecting the terms that are quadratic in small perturbations, the simplified equation of motion in the forward (x) direction is,

$$\Delta X - mg_0 cos(\theta)\theta = m\dot{u} \tag{10}$$

After using the forward Euler method to numerically solve the ODE, we observed that after an initial instability, the ODE stabilises eventually as observed in Figure 2. The MATLAB code is given in full in Figure 5, Appendix A for reference. As the damping ratio for this system is greater than one, we observe that an overdamped system does not oscillate and it returns to rest exponentially. [11].

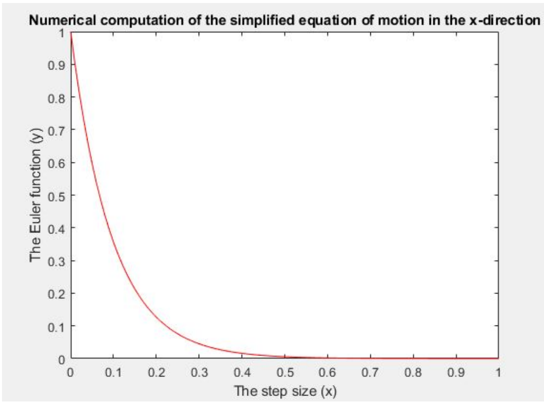

Figure 2: Numerical computation of the simplified equation of motion in the x-direction. $\Delta X = 0.01, m = 80000kg$, $g_0 = 9.81$, $\theta_0 = 0$.

2.2 Second model: 3 DOF non-linear model

We still use Newton's laws of motion to derive the equations of motion for the aircraft. As shown above we use the generalised force equations (Eq.6 and Eq. 14 - 16) and model them in MATLAB (please refer to Figure 10 and 11 in Appendix B for the complete MATLAB code) as a function of time. We try to determine how the system behaves when the time values are varied between 0 to 20 seconds. Thus the generalised equations are,

$$X - mg_0 sin\theta = m(\dot{u} + qw - rv) \tag{11}$$

$$Y = m(\dot{v} + ru - pw) \tag{12}$$

$$Z + mg_0 cos\theta = m(\dot{w} + pv - qu) \tag{13}$$

This is equal to

$$\dot{u} = \frac{X}{m} - g_0 sin\theta - qw + rv \tag{14}$$

$$\dot{v} = \frac{Y}{m} - ru + pw \tag{15}$$

$$\dot{w} = \frac{Z}{m} + g_0 cos\theta - pv + qu \tag{16}$$

This model still does not contain roll and yaw angles which would considerably change the rotation matrix above, which only incorporates the pitching angles. The first case we consider to have constant angular velocity with p, q, r all set to 0.01. The initial perturbations in the X, Y, Z space are also all set to 0.01 for the first simulation. The acceleration due to gravity is set constant to 9.81 ms^{-2}. We plot the velocity values against the time values to determine the relationship between the two parameters, to simulate aircraft motion. For all cases we take the initial velocities set to 0 ms^{-1} for u, v, w.

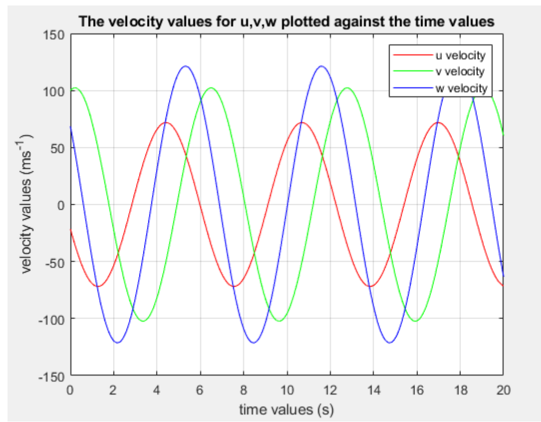

Figure 3: Numerical computation of the simplified equations of motion in the x, y and z-axes. $\Delta X, \Delta Y, \Delta Z = 0.1; p, q, r = 0.1; m = 80000kg$; g= 9.81; 0$\leq$ time $\leq$ 20 s

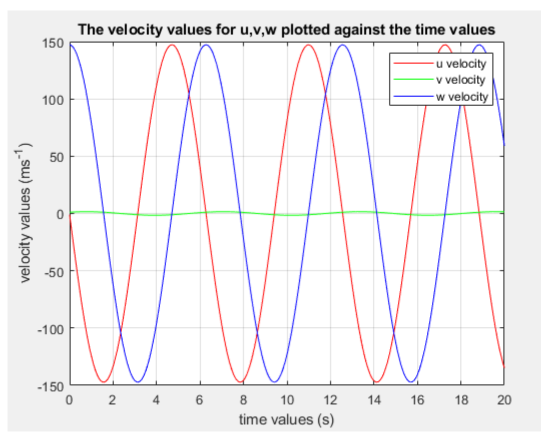

Figure 4: Numerical computation of the simplified equations of motion in the x, y and z-axes. $\Delta X, \Delta Y, \Delta Z = 0.001; p, q, r = 0.001; m = 80000kg$; g= 9.81; 0$\leq$ time $\leq$ 20 s

We observe a sinusoidal graph in both Figures 3 and 4. We observe that with reduction in the values for $\Delta X, \Delta Y$and ΔZ and for p, q, r, we see that the v velocity settles to equilibrium but that the u, w velocities display signs of phugoid mode, the nature of which will be discussed further in the text, in detail. This instability is a direct result of the simplicity of the model, which higher order analysis, we expect to be able to observe a damped oscillation, where the velocity values are quickly damped (indicating a manual manoeuvre to prevent instability during flight). To this end, we see the sideways velocity getting damped the fastest, which makes intuitive sense as the fluctuations in the x and z-axes may be chalked to wind velocities and other sources of turbulence not considered in the model.

3 Classical mechanics: Lift and drag

3.1 Third model: Euler approximation

Like the previous models we still use Newton's laws of mechanics to derive equations related to an aircraft. As we are only looking at pitch within this model we can ignore certain factors that will affect the aircraft. First there are two places in which lift occurs on an aeroplane. The first is at the centre of mass. We assume the centre of mass is in the centre of the wings as this is approximately correct. This lift only affects the altitude of the aircraft and has no relation to pitch so this can be ignored. Next is the lift that occurs at the tail of the aeroplane. This provides a rotation around the centre of mass so this is calculated. For this we take the drag force acting on the aeroplane's tail and find the rotational component of this relative to the centre of mass. First the value of drag is required. Drag is dependant on the coefficient of drag, density of the air, the square of velocity and the relative area. This area can be the area of the wings on the tail, the surface area of the aeroplane or any other measure for area. This is because whatever is chosen will be reflected within the drag coefficient as this coefficient is relative to the certain body chosen.

$$D = \frac{1}{2} \times C_d \times \rho \times V^2 \times A \tag{17}$$

Using the calculated value for drag we use trigonometry to find the lift where L_T is the lift at the tail and alpha is the angle of attack of the aeroplane.

$$L_T = D \times sin(\alpha) \tag{18}$$

However, as we are only looking at small changes in angle of attack the equation can be simplified using the small angle theorem.

$$L_T = D \times \alpha \tag{19}$$

This lift is rotational force acting on the tail. From here we can work out the torque acting around the centre of mass using the distance between the centre of mass and the tail as l.

$$\tau = L_T \times l \tag{20}$$

Using torque and the moment of inertia of the aircraft we can find its angular acceleration which in turn can be used to find the change in the angular velocity which is required for the model. To work out the moment of inertia some guesswork is required as the shape of an aeroplane is irregular. If we use the following equation and the estimate that the concentration of the mass is equally divided in two places both a quarter of the way in from the nose and the tail we find that the moment of inertia is $\frac{1}{4}ml^2$.

$$I = \sum_{i=1}^{\infty} m_i r_i^2 \tag{21}$$

Using this we can find the angular acceleration for the aeroplane which in turn can be used with the forward Euler method to model the angle of attack over a period of time.

$$\frac{d\omega}{dt} = -\frac{\tau}{I} \tag{22}$$

This equation is the first of the derivatives used within the Euler method. The second is the angular velocity of the aircraft. This value will be worked out using the above equation as the value of ω will change with each iteration.

$$\frac{d\alpha}{dt} = \omega \tag{23}$$

Using these derivatives we can perform Euler's method to analyse the change in the angle of attack for a given time period. For this we need to work out the change in both ω and α. To do this we use the following equations where t is time.

$$\Delta\omega = \frac{\omega}{dt} \times \Delta t \tag{24}$$

$$\Delta\alpha = \omega \times \Delta t \tag{25}$$

$$\omega_{n+1} = \omega_n + \Delta\omega \tag{26}$$

$$\alpha_{n+1} = \alpha_n + \Delta\alpha \tag{27}$$

For Euler's method Eq. 24 - 27 are repeated for the given time span and the values of α are recorded and then plotted against time to show how the angle of attack changes with time. Using estimated values for a Cessna 172, this is the result [4] [1].

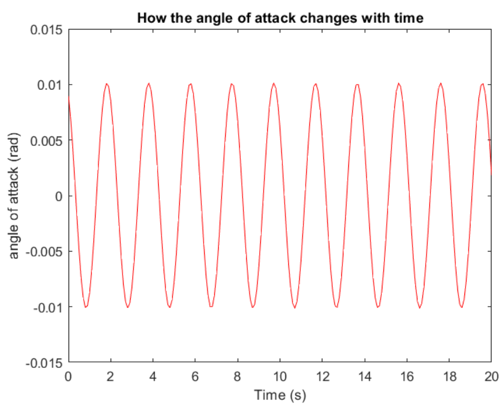

Figure 5: A graph to show how the angle of attack of a plane responds to an initial disruption.

From this graph, we can see that the simple model created shows the angle of attack oscillating over time with no change in amplitude. This means that the aircraft modelled has neutral dynamic stability as the oscillations never dampen out over time [5]. This is however only due to the simplicity of the model as a Cessna 172 has positive dynamic stability which means realistically this model should dampen and revert to the natural angle of attack of the plane. There are three factors that effect the dynamic stability of a plane and determine whether it will be positive and if so, how fast a disruption in the angle of attack will be corrected [10]. First, a centre of gravity closer to the nose of the plane generally means the plane will be more stable with respect to the aircraft's pitching moment. There will be a place on the aircraft where if the centre of gravity lies the plane will have neutral stability and any further back from this point will result in the plane being negatively dynamically stable. This is why small planes have a strict restriction on the load they can take and how it is loaded. This factor will not be considered within the model as it is assumed that the plane will have positive dynamic stability[10].

Next the position of the centre of pressure on the aircraft determines if there will be any pitching moment. The centre of pressure is the point on the aeroplane where it can be imagined the lift force is "concentrated". This is a similar concept to how centre of gravity is thought of. If the centre of pressure does not "act" in the same place as the centre of gravity there will be a consequential pitching moment. Due to the complexity of calculating the centre of gravity, this will also be excluded from the model[10].

Lastly, the elevator component on a plane controls the pitching rotations. It is possible for the elevator to be designed in such a way that it can have passive restoring capabilities which will dampen out any change in pitch. For this to occur the any extra or loss in lift from the wings (dependant on whether the plane is pitching up or down) is countered by a lift greater in magnitude from the tail due to an passive change in the angle of attack from the elevator. This passive change can be done passively using a well though design or a flight computer which corrects the path (auto-pilot). A small plane is unlikely to have such capabilities of auto-pilot so this passive change will be from a well designed tail-plane [10]. This factor will be implemented within the model.

The premise of this factor is that there is an increased magnitude for the angle of attack at the tail compared to the wings which in turn allows for a greater lift force and in turn torque to correct the plane. The amount at which the angle of attack is increased at the tail is proportional to the increase in lift from the change in rotation and the change in the position of the centre of pressure. The centre of pressure is proportional to the angle of attack and can therefore be represented by the angle of attack multiplied by a constant. The change in lift is also proportional to the angle of attack so this will also be represented by the angle of attack multiplied by a constant. Therefore we can say that the increase in angle of attack at tail of the plane is simply:

$$\Delta\alpha_T = \alpha \times C \tag{28}$$

The constant value of 'C' determines how fast the plane will revert to the cruise angle of attack and is this is therefore the dampening constant. The larger this is the faster the plane will dampen. This is to a certain degree as too large of a value will cause instability. Using a reasonable value for this constant and the same values for all other constants as before this graph is obtained.

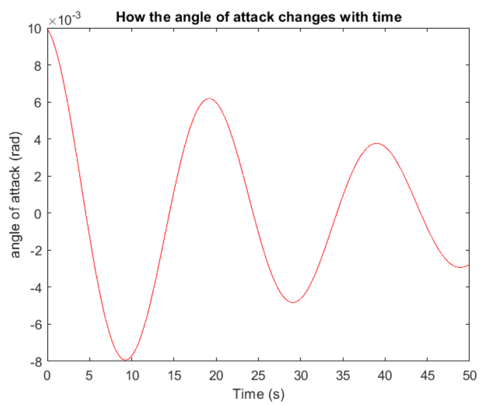

Figure 6: A graph to show how the angle of attack of a plane responds to an initial disruption with a passive elevator.

From this graph we can see that the value chosen for the dampening constant is reasonable to model a plane of this size as it is likely the plane will not immediately return to the original angle of attack whilst also not taking too long to dampen.
Next we must account for change in forward velocity. This is because the pitch and forward velocity effect each other. To start an equation for the change in velocity is required. To derive this first we look at force balancing the plane. We however do this using the plane as a frame of reference. This will give use the velocity but in the direction of the plane and not relative to earth. There are 3 forces that have a component acting parallel to the plane. Weight, drag and thrust. The weight component will be the weight multiplied by the sine of the angle of attack as this is the part that effects the velocity. The drag and the thrust act in the same line as the plane as need no translation. The thrust will be provided by the propeller of the aircraft.

$$m\frac{dV}{dt} = -D - mgsin(\alpha) + Th \tag{29}$$

From this we can divide through by mass to get the rate of change of velocity.

$$\frac{dV}{dt} = \frac{-D - mgsin(\alpha) + Th}{m} \tag{30}$$

We can again apply Euler's forward method to the model using this differential to produce a model of how the angle of attack and velocity change with time after a disturbance. From this, we get two graphs. The first shows the angle of attack against time. From this graph we can see that adding a varying velocity has not affected the rate of change of the angle of attack as the period in this graph is the same of that from the previous version of the model.

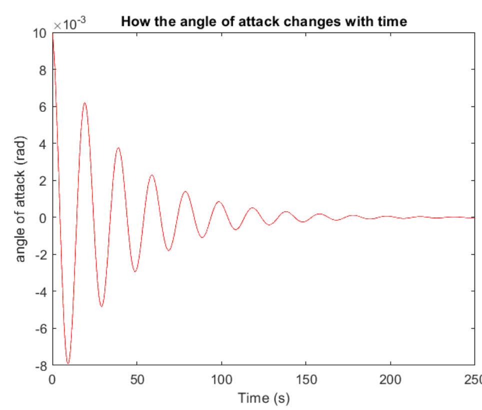

Figure 7: A graph to show how the angle of attack of a plane responds to an initial disruption with a passive elevator and a changing velocity.

This graph shows the change in velocity over time. We can see plainly from this graph that the velocity is affected by a varying angle of attack. It is also clear that it converges towards a “cruise” velocity just as the angle of attack will converge towards its original angle of attack. We can also see that the velocity graph shares the same period as the angle of attack graph. This leads me to believe that velocity is directly proportional to the angle of attack.

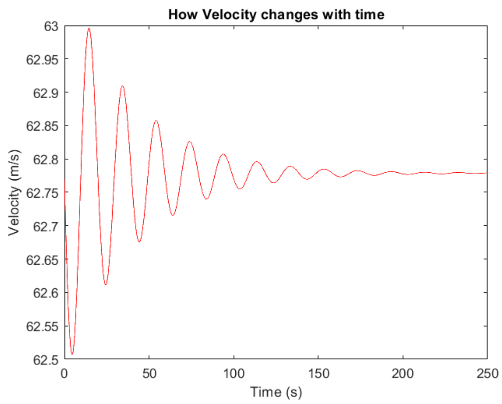

Figure 8: A graph to show how the forward velocity of a plane responds to an initial disruption with a passive elevator and a changing velocity.

This velocity is the forward velocity of the plane but in its own frame of reference. To get the velocity relative to the Earth’s frame of reference the current velocity must be multiplied by the cosine of the angle of attack.

$$V = V \times sin(\alpha) \tag{31}$$

The use of this equation in practice minutely changes the outcome of the model, so much so that the graphs look identical. This is because the offset angle of attack is so slight. When looking at a greater change in the angle of attack this would be more useful.

4 Phugoid mode

The phugoid behaviour is the rapid increase and decrease of pitch angles accompanied by fluctuating velocities which continuously makes the aircraft go “uphill” and “downhill”. The phugoid mode proceeds at a constant angle of attack and we assume that the pitch rate is very small. Thus, we approximate the behaviour of the mode by writing only the X and Z-force equations [3] where u_0 is the initial velocity of the aircraft,

$$\dot{u} = \Delta X u + \Delta X w - g_0 cos(\theta_0)\theta \tag{32}$$

$$(1 - \Delta Z)\dot{w} = \Delta Z u + \Delta Z w + (u_0 Z_q)q - g_0 sin(\theta_0)\theta \tag{33}$$

After setting $w = \dot{w} = 0$ we can write it in the matrix form,

$$\frac{d}{dt}\begin{pmatrix} u \\ \theta \end{pmatrix} = \begin{pmatrix} \Delta X & -g_0 cos\theta_0 \\ \frac{-\Delta Z}{u_0 + Z_q} & g_0 sin\theta_0 \end{pmatrix}\begin{pmatrix} u \\ \theta \end{pmatrix} \tag{34}$$

Considering level flight equilibrium and neglecting and $\theta = 0$ then,

$$\frac{d}{dt}\begin{pmatrix} u \\ \theta \end{pmatrix} = \begin{pmatrix} \Delta X & -g_0 \\ \frac{-\Delta Z}{u_0} & 0 \end{pmatrix}\begin{pmatrix} u \\ \theta \end{pmatrix} \tag{35}$$

The characteristic equation for the matrix in the equation above can be given by,

$$\lambda^2 - \Delta X \lambda - \frac{g_0}{u_0}\Delta Z = 0 \tag{36}$$

The natural frequency (ω_n)and the damping ratio (ζ) for this mode can then be given as,

$$\omega_n = \sqrt{\frac{-g_0}{u_0}\Delta Z} \tag{37}$$

$$\zeta = \frac{|\Delta X|}{2\omega_n} \tag{38}$$

Further neglecting compressibility effects we simplify (ω_n) to,

$$\omega_n = \sqrt{2}\frac{g_0}{u_0} \tag{39}$$

Using the data for a Cessna 182 Skylane, with mass 1200 kg, our simulation from our second model, gives u_0 of 71.84 (2 d.p) ms^{-1}. Thus, we determine,

$$\omega_n = \sqrt{2}\frac{9.81}{71.84} = 0.19s^{-1}$$

$$\zeta = \frac{0.01}{2 \cdot 0.19} = 0.03$$

We compare these values to $\omega_n = 0.21$ and $\zeta = 0.08$ obtained from a more rigorous analysis [9]. We observe that our simplistic analysis using the simulation under-predicts both the natural frequency by about 10 % and the damping ratio by a factor of 3/8. Nonetheless, this simplified approach gives us better intuition towards the parameters governing the phugoid mode.

5 Short-period mode

The short-period mode mainly reflects the characteristics of the aircraft's pitch rotation. Compared with the long-period mode, it has fast attenuation and high oscillation frequency. Among them, the changes in the aircraft's pitch angular velocity and angle of attack are obviously, but the changes in speed are small [16]. The short-period mode works only in the initial phase (in a few seconds) of the perturbed motion and decays quickly.

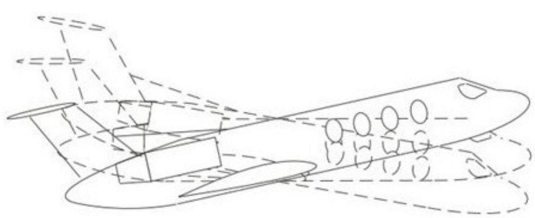

Figure 9: Short-period motion.

The speed change in the short-period motion phase is small, so the flight speed can be considered as approximately maintained [18]. Therefore, the tangential force equation in the motion can be removed. Set $\Delta v = 0$ in the rest of equations, and introduce $\frac{d\Delta\vartheta}{dt} = \omega_z$. Assume $\theta = 0$ in undisturbed motion, then equations of short-period perturbation motion can be simplified as,

$$\begin{cases} \dfrac{d\Delta\alpha}{dt} + \overline{Y}_c^{\alpha}\Delta\alpha - \omega_z = 0 \\ \\ -\overline{M}_z^{\dot{\alpha}}\frac{d\Delta\alpha}{dt} - \overline{M}_z^{\alpha}\Delta\alpha + \frac{d\omega_z}{dt} - \overline{M}_z^{\omega}\omega_z = 0 \end{cases} \tag{40}$$

Then get the characteristic equation [2] as

$$\lambda^2 + a_1\lambda + a_2 = 0 \tag{41}$$

for which

$$\lambda_{1,2} = n + i\omega = -\frac{a_1}{2} \pm \frac{\sqrt{4a_2 - a_1^2}}{2}i \tag{42}$$

$$a_1 = \overline{Y}_c^{\alpha} - \overline{M}^{\omega_z}z - \overline{M}^{\dot{\alpha}}z, a_2 = -\overline{M}_z^{\alpha} - \overline{Y}_c^{\alpha}\overline{M}_z^{\omega_z} \tag{43}$$

According to the stability criterion, the stability conditions for short-period motion are the coefficients a_1 and a_2 of the characteristic equation must be greater than zero. For regular aircraft,$\overline{Y}_c^{\alpha} > 0, \overline{M}_z^{\alpha} < 0, \overline{M}_z^{\omega} < 0$. So coefficient a_1 remain positive which satisfied the condition. Therefore, whether a_2 is greater than zero becomes the only condition for short-period modal stability.

$$-\overline{M}_z^{\alpha} - \overline{Y}^{\alpha}z\overline{M}_z^{\omega} > 0$$

The above conditions indicate that if the aircraft has static stability,$M_z^{\alpha} < 0$, then the conditions are met, and the short-term motion of the aircraft is stable. If $M_z^{\alpha} > 0$, when $-\overline{M}_z^{\alpha} - \overline{Y}^{\alpha}z\overline{M}_z^{\omega} > 0$, the motion is still able to remain stable.

6 Discussion and Conclusion

The first and second models are formulated entirely on Newton's laws of motion. The first model takes a simplistic Euler method approximation to the solution, and we observe the system getting completely damped. The second model uses MATLAB's powerful ODE solver to get a sinusoidal curve where the sideways velocity then goes to zero which makes intuitive sense as explained in the model discussion, yet the instability in the model remains, which is then solved in the third model, using estimated damping constants.
The instability of the system in the second model can be thought of as phugoid behaviour. The phugoid mode makes the aircraft undergo rapid changes in pitch angles and velocities, leading to instability in the system. This instability can be assessed using the phugoid mode equations, using the damped natural frequency and the damping ratio as parameters. We find that the simulation under-predicts both the natural frequency and the ratio by 10% and a factor of 3/8 respectively, yet the simulation can

be considered to be useful by its simplistic application to problems.
The third model created only uses drag forces that act on the tail of the plane when calculating the pitching moment of the aeroplane. This is because this is the bulk of the contribution towards the pitching moment and drag can be used to represent both the drag and the lift forces acting on the tail. This makes the model a lot more simplistic whilst not losing a great deal of accuracy and authenticity. However, when looking at bigger angles of attack and greater changes in angles of attack this model will be too simplistic to accurately predict the dynamics of a plane. In a more complex model, the centre of pressure for the aircraft would be determined as this is a large part in how a well-designed elevator passively dampens the oscillations of a plane. However, for a simplistic model, a carefully chose constant along with the angle of attack can provide a mediocre representation of the centre of pressure. One large flaw within the third model is the lack of information on hand when creating it. This leaves large room for error as a lot of variables, for example, the drag coefficient, are estimated or changed in order to give a better functioning model. Despite the simplicity of the model the output graphs are reasonable and whilst they may be inaccurate, they do represent how different forces that act on a plane are affected by the change in the angle of attack and they effect change there after.
The combination of the two models will give a good understanding of the dynamics of an aeroplane and how these dynamics will change with different aircraft as variables are changed.

References

[1] U.s. standard atmosphere. *Engineering ToolBox.*

[2] Benwu Zheng, H. X. (1990). *Aerodynamics of aircraft with automaton.* Aviation Industry Press.

[3] Caughey, D. A. (2011). Introduction to aircraft stability and control course notes for m&ae 5070. *Sibley School of Mechanical & Aerospace Engineering Cornell University.*

[4] Cessna (2011). Cessna 172. *Cessna Aircraft Company.*

[5] Cutler, C. (2015). The 3 types of static and dynamic aircraft stability. *Online Flight Training Courses and CFI Tools.*

[6] Duke, E. L., Antoniewicz, R. F., and Krambeer, K. D. (1988). Derivation and definition of a linear aircraft model.

[7] Etkin, B. (1972). Dynamics of atmospheric flight, john wiley and sons. *Inc, New York, London, Sydney.*

[8] Ferrarese, G. (2018). Natural motion of a multi-rotor aircraft. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 232(2):280–289.

[9] Foster, T. and Bowman, J. (2005). Dynamic stability and handling qualities of small unmanned- aerial vehicles. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics.

[10] Groh, R. (2018). Control and stability of aircraft. *Aerospace Engineering Blog.*

[11] Inman, D. J. and Singh, R. C. (1994). *Engineering vibration*, volume 3. Prentice Hall Englewood Cliffs, NJ.

[12] Johnson, E. and Mishra, S. (2002). Flight simulation for the development of an experimental uav. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, page 4975.

[13] L'Afflitto, A. (2017). Equations of motion of an aircraft. In *A Mathematical Perspective on Flight Dynamics and Control*, pages 35–64. Springer International Publishing.

[14] Shy, K. S., Hageman, J. J., and Le, J. H. (2002). *The role of aircraft simulation in improving flight safety through control training.* National Aeronautics and Space Administration, Dryden Flight Research Center.

[15] Svoboda, M., Marek, J., and Heckenbergerová, J. (2014). Estimation of angles yaw, pitch and roll in a special transformation problem. In *Nostradamus 2014: Prediction,*

Modeling and Analysis of Complex Systems, pages 393–400. Springer International Publishing.

[16] Tseng, J.-B. and Lan, C. E. (1988). Calculation of aerodynamic characteristics of airplane configurations at high angles of attack.

[17] Wu, J. H., Zhang, Y. L., and Sun, M. (2009). Hovering of model insects: simulation by coupling equations of motion with navier–stokes equations. *Journal of Experimental Biology*, 212(20):3313–3329.

[18] Zhenping, F., Wanchun, C., Shuguang, Z., et al. (2005). “aircraft flight dynamics.

Appendix A Preliminary model: Euler approximation

```
h=0.01; % step size
g = 9.81;
N=100; % number of steps
y(1)=1; %initial condition
for n=1:N
y(n+1)= y(n)+h*((0.01/80000) - g.*cos(0).*y(n));
x(n+1)=n*h;
end
plot(x,y,'r')
```

Figure 10: MATLAB code for the Euler method approximation for the forward equation of motion for the aircraft.

Appendix B Second model: Non-linear model using ode45 for stiff ODEs

```
function dxdt = vanderpolaircraft(~, x, theta)
%Authors: Vedang Joshi, Joshua Lynch, Vincent Peng, Elfa Ren

%Inputs:
% x = the vector that contains u, v, w. x(1) points to 'u', x(2) points
% to 'v', x(3) points to 'w'.
% theta = the parameter to be changed in the three equations of motion.
% It is the pitch angle for the aircraft.


g = 9.81; %the acceleration due to gravity on earth

%The following quantities may be changed:

m = 80000;     %the mass of the aircraft
deltaX = 0.01; %small change in the forward displacement of the aircraft
deltaY = 0.01; %small change in the sideways displacement of the aircraft
deltaZ = 0.01; %small change in the downward displacement of the aircraft

deltap = 0.01; %small change in the rotation (x-axis) of the aircraft
deltaq = 0.01; %small change in the rotation (y-axis) of the aircraft
deltar = 0.01; %small change in the rotation (z-axis) of the aircraft

% Equations of motion as derived in the workings
udot = deltaX./m - g.*sin(theta) - deltaq.*x(3) + deltar.*x(2);
vdot = deltaY./m - deltar.*x(1) + deltap.*x(3);
wdot = deltaZ./m + g.*cos(theta) - deltap.*x(2) + deltaq.*x(1);
dxdt = [udot; vdot; wdot];

end
```

Figure 11: MATLAB code for the Van-der-Pol equation used to solve for the RHS of the generalised equations of motion for the aircraft.

```matlab
function [theta, x] = odeapplication(tMin, tMax, uInit, vInit, wInit)
%Authors: Vedang Joshi, Joshua Lynch, Vincent Peng, Elfa Ren
%Inputs:
%tMax = the final value of the time duration (minutes)
%tMin = the initial value of the time duration (minutes)
%uInit = initial value of u i.e. u(0) = uInit
%vInit = initial value of v i.e. v(0) = vInit
%wInit = initial value of w i.e. w(0) = wInit

%initialise arrays to hold calculated u, v, w values
uArray = [];
vArray = [];
wArray = [];

%use the Van-der-Pol function derived in the vanderpolaircraft.m file
%to implement in the ode45 functionality in MATLAB
for theta = -10:0.01:10
    [~, x] = ode45(@(t,x) vanderpolaircraft(t, x, theta), [tMin tMax], ...
                                                [uInit vInit wInit]);
    uArray = [uArray x(end,1)]; %#ok<AGROW>
    vArray = [vArray x(end,2)]; %#ok<AGROW>
    wArray = [wArray x(end,3)]; %#ok<AGROW>

end

% plot the graphs of theta against the values obtained for u, v, w, stored
% in the respective arrays
theta = linspace(-10,10,length(uArray));
plot(theta, uArray, '-r');
xlabel('theta values (°)');
ylabel('velocity values (ms^{-1})');
title('The velocity values for u,v,w plotted against the pitch angle values')

grid on
hold on
theta = linspace(-10,10,length(vArray));
plot(theta, vArray, '-g');
theta = linspace(-10,10,length(wArray));
plot(theta, wArray, '-b');
hold off

legend('u velocity', 'v velocity', 'w velocity');
end
```

Figure 12: MATLAB code for the ode45 method approximation for the velocities in the x, y and z-axes for the aircraft.

Appendix C Third model: Using Euler's approximation

```
function D = drag(Cd, p, A, V, a)

D = Cd*p*A*(V^2)*0.5*(1.01-cos(a));

end
```

Figure 13: MATLAB code for the approximation of the drag force.

```
function T = torque(l, D, a)

T = l*D*sin(a);

end
```

Figure 14: MATLAB code for the approximation of the torque.

```
function dwdt = angular_acceleration(T, I)

dwdt = -T/I;

end
```

Figure 15: MATLAB code for the approximation of the angular acceleration.

```
function aircraft_model(t0, delt_t, t_final, a0)
close all;
%Velocity stays constant
m = 1111; %mass [kg]
l = 6; %length from wings to tail in [m]
I = 0.25*m*(l^2); %moment of inertia [kgm2]
V = 62.78; %velocity [ms-2]
Cd = 1; %drag coefficient
p = 0.5258; %air density at 8,000 feet [kgm-2]
A = 16.2; %area of wings [m2]
a = a0; %initial disturbed angle of attack [rad]
w = 0; %angluar velocity [rads-1]
%creating arrays to plot
maxsize = length(t0:delt_t:t_final);
a_array = zeros(1, maxsize);
t_array = zeros(1, maxsize);
it = 0; %initialising a iteration counter
%loop to iterate through all values of time
for t=t0:delt_t:t_final
    it = it + 1;
    D = drag(Cd, p, A, V, a);
    T = torque(l, D, a);
    delt_w = angular_acceleration(T, I)*delt_t;
    w = w + delt_w; %creating new angular velocit
    delt_a = w*delt_t;
    a = a + delt_a; %creating new angle
    a_array(it) = a; %adding values to array
    t_array(it) = t; %adding values to array
end
plot(t_array, a_array, '-r'); xlabel('Time (s)');
    ylabel('angle of attack (rad)');...
    title('How the angle of attack changes with t
end
```

Figure 16: MATLAB code for the initial model's approximation of the change of angle of attack with time using Euler's forward method.

```
function aT = passive_elevator(a)

C = 0.005;

aT = -a*C;

end
```
Figure 17: MATLAB code for the approximation of a passive elevator's change in angle of attack.

```
function aircraft_model_positive(t0, delt_t, t_final, a0)
close all;
%list of all constant variables needed
m = 1111; %mass [kg]
l = 6; %length from wings to tail in [m]
I = 0.25*m*(l^2); %moment of inertia [kgm2]
V = 62.78; %velocity [ms-2]
Cd = 1; %drag coefficient
p = 0.5258; %air density at 8,000 feet [kgm-2]
A = 16.2; %area of wings [m2]
a = a0; %initial disturbed angle of attack [rad]
w = 0; %angluar velocity [rads-1]
%creating arrays to plot
maxsize = length(t0:delt_t:t_final);
a_array = zeros(1, maxsize);
t_array = zeros(1, maxsize);
it = 0;%initialising a iteration counter
%loop to iterate through all values of time
for t=t0:delt_t:t_final
    it = it + 1;
    D = drag(Cd, p, A, V, a);
    T = torque(l, D, a);
    delt_w = angular_acceleration(T, I)*delt_t;
    w = w + delt_w; %creating new angular velocity
    delt_a = w*delt_t;
    a = a + delt_a + passive_elevator(a); %creating new angle
    a_array(it) = a; %adding values to array
    t_array(it) = t; %adding values to array
end
plot(t_array, a_array, '-r'); xlabel('Time (s)'); ...
    ylabel('angle of attack (rad)');...
    title('How the angle of attack changes with time');
end
```

Figure 18: MATLAB code for the improved model's approximation of the change of angle of attack with time using Euler's forward method.

```
function dVdt = para_velocity(Cd, p, A, V, m, a, g, Th)

dVdt = (-m*g*sin(a) - drag(Cd, p, A, V, a) + Th)/(m);

end
```

Figure 19: MATLAB code for the approximation of the rate of change of an aircraft's velocity.

```
function aircraft_model_change_v(t0, delt_t, t_final, a0)
close all;
m = 1111; %mass [kg]
l = 6; %length from wings to tail in [m]
I = 0.25*m*(l^2); %moment of inertia [kgm2]
V = 62.78; %velocity [ms-2]
Cd = 1; %drag coefficient
p = 0.5258; %air density at 8,000 feet [kgm-2]
A = 16.2; %area of wings [m2]
a = a0; %initial disturbed angle of attack [rad]
w = 0; %angluar velocity [rads-1]
g = 9.81; %gravitational constant [ms-2]
Th = 167.8603883; %thrust provided from propeller [kgms-2]
%creating arrays to plot
maxsize = length(t0:delt_t:t_final);
a_array = zeros(1, maxsize);
t_array = zeros(1, maxsize);
it = 0;%initialising a iteration counter
%loop to iterate through all values of time
for t=t0:delt_t:t_final
    it = it + 1;
    D = drag(Cd, p, A, V, a);
    T = torque(l, D, a);
    delt_w = angular_acceleration(T, I)*delt_t;
    w = w + delt_w;
    delt_a = w*delt_t;
    delt_V = para_velocity(Cd, p, A, V, m, a, g, Th)*delt_t;
    a = a + delt_a + passive_elevator(a);
    V = V + delt_V;
    a_array(it) = V;
    t_array(it) = t;
end
plot(t_array, a_array, '-r'); xlabel('Time (s)'); ylabel('Velocity (m/s)');...
    title('How Velocity changes with time');
end
```

Figure 20: MATLAB code for the final model's approximation of the change of angle of attack with time using Euler's forward method.

```
function HV = horizontal_velocity(V, a)

HV = V * cos(a);

end
```

Figure 21: MATLAB code for the approximation of the horizontal velocity of an aircraft.