

Lecture D3 - Equations of Motion in Cartesian Coordinates (2DOF Aircraft Model)

Newton's second law

$$\mathbf{F} = m\mathbf{a}$$

is a vector equation that relates the magnitude and direction of the force vector, to the magnitude and direction of the acceleration vector. In the previous lecture we derived expressions for the acceleration vector expressed in cartesian coordinates. This expressions can now be used in Newton's second law, to produce the equations of motion expressed in cartesian coordinates.

Equations of Motion in Cartesian Coordinates

In rectangular cartesian coordinates, xyz , we write $\mathbf{F} = F_x\mathbf{i} + F_y\mathbf{j} + F_z\mathbf{k}$ and $\mathbf{a} = a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k}$. Thus, we have, in component form

$$\begin{aligned} F_x &= ma_x = m\ddot{x} \\ F_y &= ma_y = m\ddot{y} \\ F_z &= ma_z = m\ddot{z} . \end{aligned} \tag{1}$$

This system of Ordinary Differential Equations (ODE's) must be augmented with appropriate initial conditions. Typically we will know the initial position \mathbf{r}_0 and velocity \mathbf{v}_0 . Thus, $\mathbf{r}(0) = x(0)\mathbf{i} + y(0)\mathbf{j} + z(0)\mathbf{k} = \mathbf{r}_0$, and $\mathbf{v}(0) = \dot{x}(0)\mathbf{i} + \dot{y}(0)\mathbf{j} + \dot{z}(0)\mathbf{k} = \mathbf{v}_0$.

In general, forces on a body depend in general on its position \mathbf{r} , its velocity \mathbf{v} and, sometimes, explicitly on the time t (e.g. the control inputs on an aircraft). In addition, they may also depend on the position and velocity of other neighboring bodies or particles. In this situation, we may need to solve for the motion of all the bodies involved simultaneously. Fortunately, one can often make the approximation that the effect of the surrounding bodies is small and hence simplify the problem considerably.

Analytical Integration – Some General Cases

There are only a few instances where the equations of motion can be solved analytically. In the general cases numerical integration is required. Among the special situations where analytical solution is possible, there are the problems where \mathbf{F} is either constant or depends on t only. In such cases we can integrate the above

equations independently to compute the components of the velocity vector,

$$v_x(t) = v_x(0) + \frac{1}{m} \int_0^t F_x(t) dt, \quad v_y(t) = v_y(0) + \frac{1}{m} \int_0^t F_y(t) dt, \quad v_z(t) = v_z(0) + \frac{1}{m} \int_0^t F_z(t) dt,$$

and the position,

$$x(t) = x(0) + \frac{1}{m} \int_0^t v_x(t) dt, \quad y(t) = y(0) + \frac{1}{m} \int_0^t v_y(t) dt, \quad z(t) = z(0) + \frac{1}{m} \int_0^t v_z(t) dt.$$

There are also one-dimensional cases in which the equations of motion can be integrated analytically (**see review on rectilinear motion**). In some other situations, it is possible to partially solve these equations analytically. For instance, if the forces can be derived from a potential, we might be able to write expressions, e.g. conservation of total energy, that are satisfied by the motion even though we may not know exactly know what the motion is.

Numerical Integration

The most general way of solving the equations of motion is by numerical integration. In this case we do not compute the solution of the problem but an approximation to it. It is typically useful to work only with first order equations, so we write,

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{z} &= v_z \\ \dot{v}_x &= \frac{1}{m} F_x(x, y, z, v_x, v_y, v_z, t) \\ \dot{v}_y &= \frac{1}{m} F_y(x, y, z, v_x, v_y, v_z, t) \\ \dot{v}_z &= \frac{1}{m} F_z(x, y, z, v_x, v_y, v_z, t) \end{aligned}$$

This is a set of 6 first order ODE's, with initial conditions

$$x(0) = x_0, \quad y(0) = y_0, \quad z(0) = z_0, \quad v_x(0) = v_{x0}, \quad v_y(0) = v_{y0}, \quad v_z(0) = v_{z0}.$$

The above set of equations is sometimes written in shorthand notation as

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, t), \quad \text{with} \quad \mathbf{X}(0) = \mathbf{X}_0 \tag{2}$$

where $\mathbf{X} = (x, y, z, v_x, v_y, v_z)^T$, $\mathbf{F} = (v_x, v_y, v_z, F_x/m, F_y/m, F_z/m)^T$, and $\mathbf{X}_0 = (x_0, y_0, z_0, v_{x0}, v_{y0}, v_{z0})^T$.

The array \mathbf{X} is usually referred to as the “*vector of state variables*”, even though it is not truly a vector. It should not be confused with a physical vector such as the velocity or acceleration. Note that the components of \mathbf{X} do not even have the same units.

Fortunately, the theory for numerically solving systems of ordinary differential equations, is very mature and good algorithms are available. Here we use a simple “Forward Euler Method” for illustration. Consider a time increment Δt (this Δt should be “small enough” so that the computed results are accurate, but on the

other hand, we will see that the smaller the timestep the more time steps we will have to compute). The idea is to start from the initial condition, and compute the solution at times Δt , $2\Delta t$, $3\Delta t$, etc. Suppose that we have already progressed along the trajectory to time t , when the components of $\mathbf{X}(t)$ are all known (this could clearly be $t = 0$ because of the initial condition). We then update all these quantities to the next discrete time $t + \Delta t$, by

$$\mathbf{X}(t + \Delta t) = \mathbf{X}(t) + \Delta t \mathbf{F}(\mathbf{X}(t), t)$$

Note

Solving ODE's with MATLAB®

The task of integrating a systems of ODE's is particularly simple if we use a computer package such as MATLAB. There exist a number of solution algorithms that one can choose from, but for our purposes almost any of them will work. The MATLAB syntax is typically of the form

$$[\mathbf{t}, \mathbf{X}] = \text{solver}(\text{odefun}, \text{tspan}, \mathbf{x0}) .$$

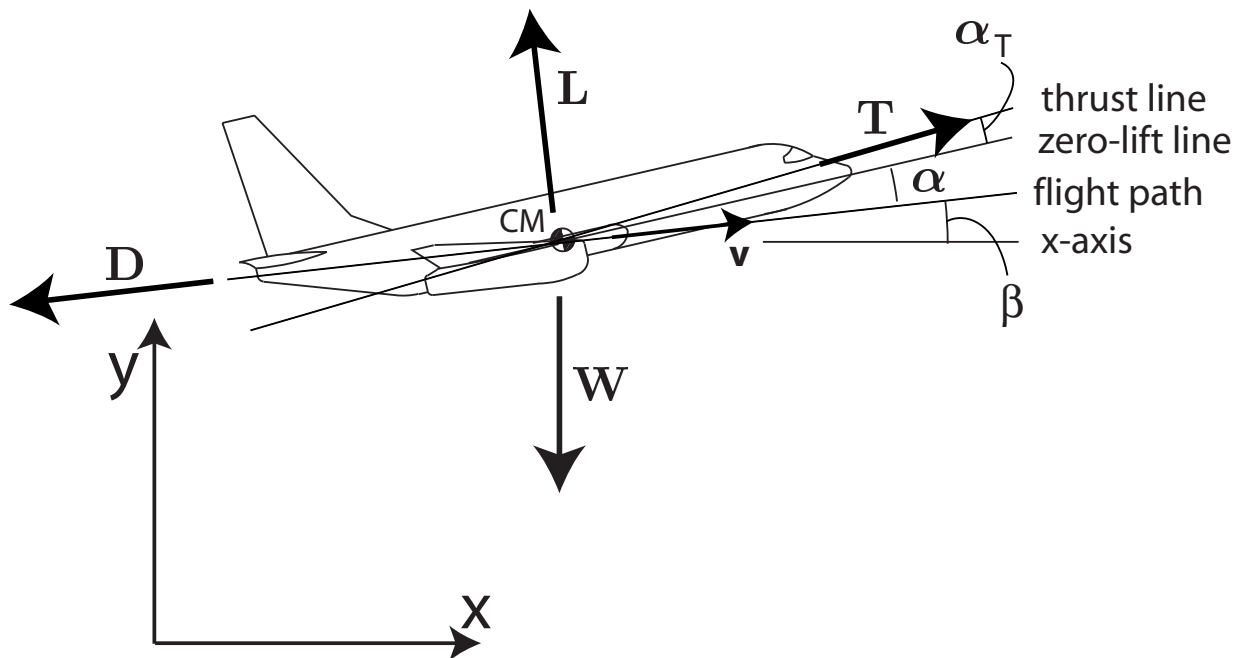
The returned arrays \mathbf{t} and \mathbf{X} contain the vector of time points and the solution, \mathbf{X} at these time points, respectively. A function, `odefun`, that calculates \mathbf{F} , from \mathbf{X} , and t , must be provided, as well as a vector specifying the interval of integration `tspan = [t0, tf]`, and the initial value of \mathbf{X} , `x0`. In order to set the parameters for the solution algorithm one can use the MATLAB function `odeset`.

See <http://www.mathworks.com/access/helpdesk/help/techdoc/ref/ode45.html> for details as well as the lab 1 handout.

2 Degrees-of-Freedom (2DOF) Aircraft Model

In this section we will develop a simple model to simulate the flight of an aircraft (or spacecraft). The aircraft is treated as a particle. Therefore, we assume that all its mass is concentrated at a point located at the aircraft's center of mass. In addition, we will assume that we have perfect attitude control. That is, we will decouple the motion of the center of mass with the rotation of the aircraft about the center of mass. For the later, we will assume that we have a control system that allows us to prescribe the angle of attach of the aircraft. In reality this is clearly not possible and any moment applied to the aircraft, through the control surfaces, to change its attitude will take a certain time to be effective. A more detailed model in which the aircraft is treated as a rigid body, with rotational inertia, will be considered later on in the course. Although approximate, the current model is found to be useful for many applications. In addition, we will assume that the aircraft flies in a vertical plane. The extension to three dimensions, although not hard will make things too complicated at this point. Therefore, the two degrees of freedom of the aircraft are the horizontal and vertical positions, x and y . In order to be able to write the equations of motion we need to look at the forces that act on the aircraft.

With reference to the figure below, we define the following angles. Let β , be the angle that the velocity vector (tangent to the flight path) makes with the x -axis. Thus, we will have $\tan \beta = v_y/v_x$. Let α be the angle of attack of the aircraft relative to velocity vector. As we will see, the angle of attack is relevant, as it determines the lift (and drag) for a given velocity and altitude. The angle of attack, is defined so that the lift generated when $\alpha = 0$, is equal to zero. In addition, we define the angle α_T as the angle that the thrust vector T makes with the zero-lift line. In most cases, this angle will be constant, but in some situations (e.g. thrust vectoring in aircraft or rockets) it can be an input, or control, variable. All the angles are positive as shown in the diagram.



The aerodynamic forces, are resolved into a lift component \mathbf{L} , orthogonal to the velocity, and a drag component \mathbf{D} , parallel to the velocity. In addition, we will have, the weight of the vehicle, \mathbf{W} , and the thrust \mathbf{T} , due to the propulsive system. When these forces are referred to the center of mass of the aircraft, we may have an additional pitching moment \mathbf{M} . For this aircraft model, we will assume that the control surface guarantees that the pitching moment about the center of mass is always zero.

Forces on the Aircraft

For a standard wing-tail configuration aircraft we have the following:

Lift

Most of the aircraft's lift is generated by the wing and horizontal stabilizer. For a given angle of attack α , the lift is proportional to the relative wind's dynamic pressure, $q = (1/2)\rho v^2$, and to the wing's area S ,

$$L = \frac{1}{2}\rho v^2 S C_L. \quad (3)$$

Here, C_L , is the non-dimensional lift coefficient, which is a function of the shape, and the angle of attack α . The variation of C_L with the angle of attack is linear, up to a maximum value $C_{L_{max}}$, of the order of 1.5 – 2.0. Thus, we have

$$C_L = a\alpha , \quad (4)$$

where $a = \partial C_L / \partial \alpha$. For an idealized flat plate wing, $a = 2\pi \text{ rad}^{-1}$, and more generally for finite wings $a = 5 \text{ rad}^{-1}$ to $a = 6 \text{ rad}^{-1}$. Note that for $\alpha = 0$, $C_L = 0$, which implies that we are assuming that $\alpha = 0$ corresponds to the zero-lift line.

Pitching Moment

The lift force, \mathbf{L} , is the result of integrating a distributed force over the surface of the wing. This distributed force is equivalent to \mathbf{L} , acting through a particular point called the center of pressure (CP). For this model, we will assume that through proper control of the flaps and tail, the pitching moment about the center of mass is zero. Moreover, we shall assume that any attitude changes commanded by the control system are instantaneous.

Drag

The drag force can also be expressed as a function of a non-dimensional drag coefficient, C_D , as

$$D = \frac{1}{2} \rho v^2 S C_D. \quad (5)$$

The drag coefficient in general has a complicated dependence on α and C_L . But commonly it is assumed to have two components, one which is independent of the lift, and one which increases quadratically with the lift,

$$C_D = C_{D0} + K C_L^2 . \quad (6)$$

Typical values for C_{D0} for the whole aircraft, are of the order of 0.003 to 0.02. An important ratio used to determine the aircraft's performance is $L/D = C_L/C_D$, which is typically of the order of 10 – 20 for most subsonic aircraft, and up to 60 for some sailplanes.

Thrust

The thrust generated by the engine is a force directed at an angle α_T measured with respect to the zero-lift line. In general, both the magnitude of T , and its direction α_T , are inputs to the model.

Weight

Typically we will simple have $W = mg$ and it will be directed along the negative y -axis. Note, however, that we are interested in spacecraft in orbit, the weight might be directed to a fixed point, e.g. the earth center, and may be a function of the height.

Equation of Motion

Written in cartesian coordinates, the equations of motion are

$$\begin{aligned} ma_x &= T \cos(\beta + \alpha + \alpha_T) - D \cos \beta - L \sin \beta \\ ma_y &= L \cos \beta + T \sin(\beta + \alpha + \alpha_T) - D \sin \beta - W \end{aligned} \quad (7)$$

State Space Form

The equations of motion can be written in the form of equation (2) by introducing the velocities as variables and taking

$$\mathbf{X} = \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} v_x \\ v_y \\ (T/m) \cos(\beta + \alpha + \alpha_T) - (D/m) \cos \beta - (L/m) \sin \beta \\ (L/m) \cos \beta + (T/m) \sin(\beta + \alpha + \alpha_T) - (D/m) \sin \beta - (W/m) \end{pmatrix} \quad (8)$$

where, T , α and α_T are inputs to the model which will typically be a function of time and perhaps \mathbf{X} through the control system. Recall that $\beta = \tan^{-1}(v_y/v_x)$, and L is by expressions (3), (4) and D , by expressions (5), (6), and (4).

ADDITIONAL READING

J.L. Meriam and L.G. Kraige, *Engineering Mechanics, DYNAMICS*, 5th Edition

3/3, 3/4, 3/5 (rectangular coordinates only)