Solar-powered Unmanned Aerial Vehicle (SUAV) PROTOTYPE Flight Model

This document outlines the flight model implemented into the prototype system and the method used to introduce uncertainties in wind and cloud cover.

0.1 Flight Model

0.1.1 Flight dynamics

All aerial vehicles have six degrees of freedom, as shown in Figure 0.1.

Developing a comprehensive flight model with six degrees of freedom (d.o.f.) is complex when accounting for rotational forces in addition to linear forces about each axis. As the project focuses on developing the mission planning system, a compromise between model simplicity and accuracy was made by neglecting any rotational forces about each axis. This removes a layer of complexity by removing the need to evaluate the angular velocities and accelerations about each axis.

Moreover, flight dynamics are dependent on relative velocities between the ground, wind and the vehicle itself. Consider Figure 0.2 with a fixed frame of reference attached to the ground. Essentially, in the presence of wind ($\underline{v}_G^W$), the air velocity is the velocity of the plane relative to the wind ($\underline{v}_W^P$), while the ground velocity is the velocity of the plane relative to the ground ($\underline{v}_G^P$). This concept is pivotal because the flight forces are evaluated from the air velocity, while the actual ground covered is governed by the ground velocity. In the absence of wind, the air and ground

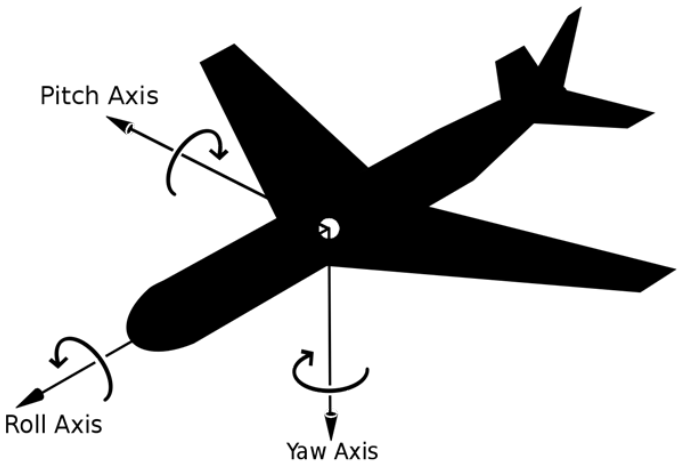


Figure 0.1: Degrees of freedom for an aerial vehicle [1].

velocities are equal. In other words,

$$\underline{v}_G^P = \underline{v}_W^P = \begin{bmatrix} \dot{x}_{gr} \\ \dot{y}_{gr} \\ \dot{z}_{gr} \end{bmatrix}$$

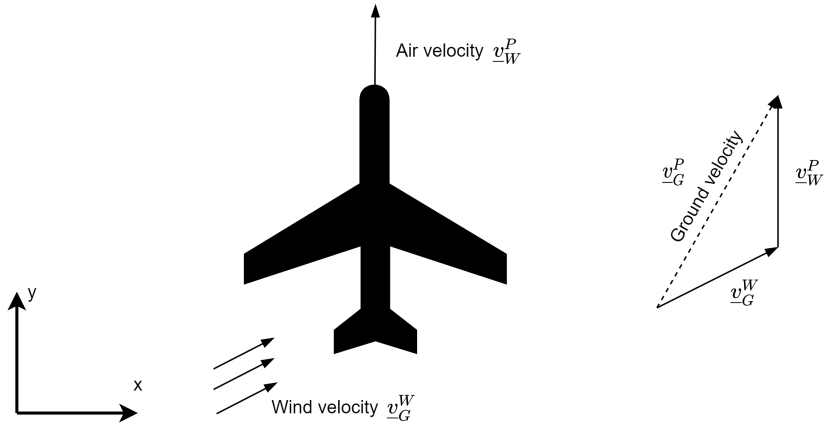


Figure 0.2: Wind, ground and air velocities - co-ordinate frame attached to ground

Consider an SUAV mid-flight with velocities in vertical (z_{gr}) and horizontal (y_{gr}, x_{gr}) directions relative to the ground, as shown in Figure 0.3.

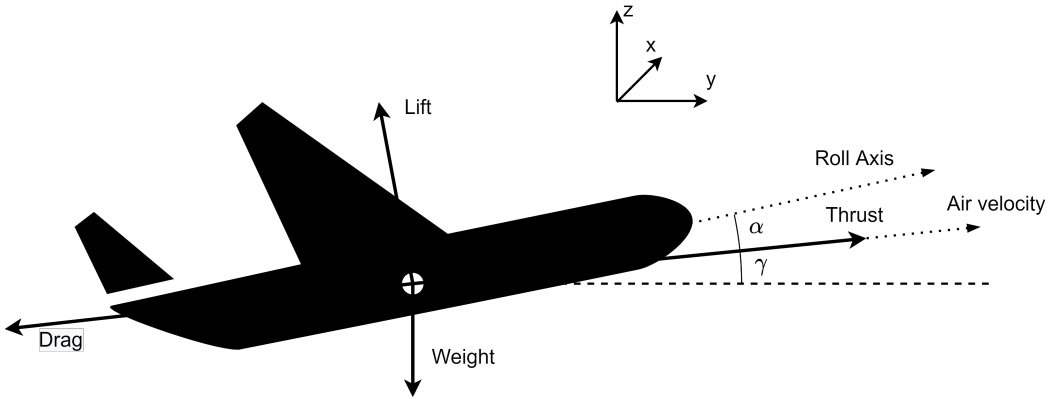


Figure 0.3: Side view of an SUAV mid-flight - co-ordinate frame attached to ground

The thrust, T, and drag, D, experienced by the SUAV acts in the direction of the air velocity, v_{air}. Additionally, lift, L, acts perpendicularly to v_{air}, and the weight, W, always acts downwards to the ground. They are defined as:

$$T = \frac{\eta_{prop} P_{prop}}{v_{air}} \tag{1}$$

$$D = \frac{1}{2}\rho A C_D v_{air}^2 \tag{2}$$

$$L = \frac{1}{2}\rho A C_L v_{air}^2 \tag{3}$$

$$W = mg \tag{4}$$

where η_{prop} is the efficiency of the UAV propellers, P_{prop} is the propelling power, ρ is density of air, A is the wing area of the UAV, m is the mass of the UAV, and g is the gravitational constant. The drag co-efficient, C_D, and lift co-efficient, C_L are dependent on the angle of attack, α.

The climb angle is a function of the directional air velocities:

$$\gamma = tan^{-1}\left(\frac{\dot{z}_{air}}{v_{air,hor}}\right) \tag{5}$$

where the horizontal air velocity, $v_{air,hor}$ is defined as:

$$v_{air,hor} = \sqrt{\dot{x}_{air}^2 + \dot{y}_{air}^2} = v_{air} cos(\gamma)$$

To determine the ground vertical acceleration, $\ddot{z}_{gr}$, Newton's second law is applied to the vertical

components of the forces:

$$\ddot{z}_{gr} = \frac{1}{m}[(T - D)sin(\gamma) + Lcos(\gamma) - W] \tag{6}$$

To determine the ground horizontal accelerations, $\ddot{x}_{gr}$ and $\ddot{y}_{gr}$, the horizontal net force, F_{hor}, as shown in Figure 0.4, is required, where:

$$F_{hor} = (T - D)cos(\gamma) - Lsin(\gamma)$$

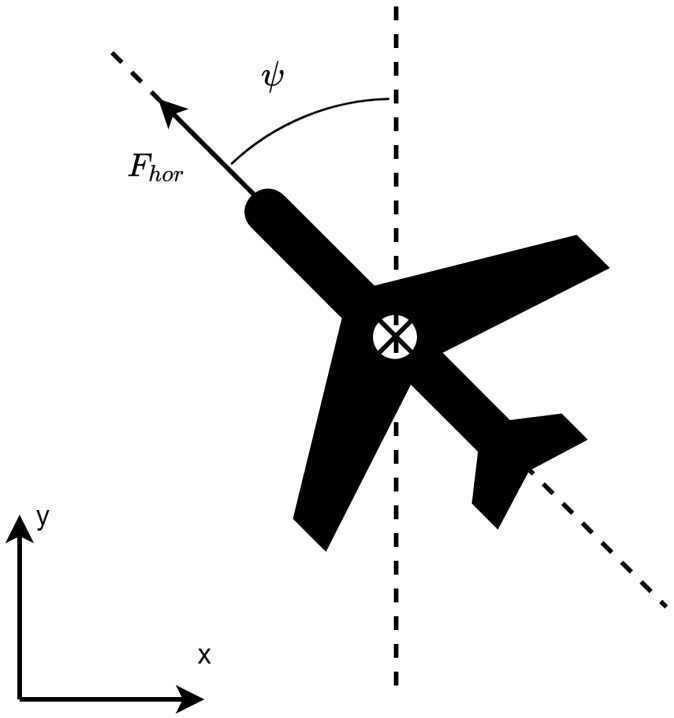


Figure 0.4: Top view of an UAV mid-flight - co-ordinate frame attached to ground with a yaw angle (ψ).

Hence:

$$\ddot{x}_{gr} = -\frac{F_{hor}}{m}sin(\psi) \tag{7}$$

$$\ddot{y}_{gr} = \frac{F_{hor}}{m}cos(\psi) \tag{8}$$

where the yaw angle is defined by:

$$\psi = tan^{-1}\left(\frac{\dot{x}_{air}}{\dot{y}_{air}}\right) \tag{9}$$

Eq. 3.9 requires the assumption that the SUAV always travels forward along its roll axis, i.e. the nose is pointing in the direction of its air velocity when viewing from the top. This is the case when the vertical stabiliser and rudder of a plane (Figure 0.5) perfectly maintains the plane flying straight, and prevents any side-to-side motion along its pitch axis.

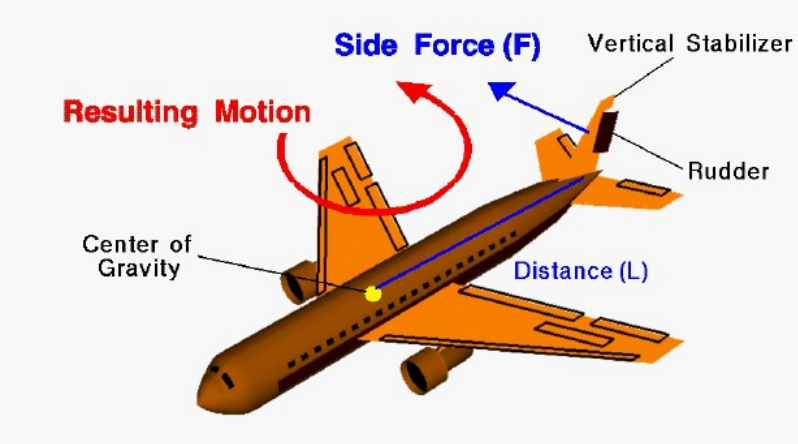


Figure 0.5: Vertical stabiliser and effect of manipulating the rudder [2].

As the air and ground velocities are equal in the absence of wind, the velocities and displacements, $\underline{d}_G^P$ can be evaluated from a set of initial displacements and velocities by directly integrating the acceleration terms with respect to time:

$$\dot{\underline{v}}_G^P = \begin{bmatrix} \ddot{x}_{gr} \\ \ddot{y}_{gr} \\ \ddot{z}_{gr} \end{bmatrix}$$

$$\underline{v}_G^P = \int_{t_1}^{t_2} \dot{\underline{v}}_G^P dt = \begin{bmatrix} \dot{x}_{gr} \\ \dot{y}_{gr} \\ \dot{z}_{gr} \end{bmatrix} \tag{10}$$

$$\underline{d}_G^P = \int_{t_1}^{t_2} \underline{v}_G^P dt = \begin{bmatrix} x_{gr} \\ y_{gr} \\ z_{gr} \end{bmatrix} \tag{11}$$

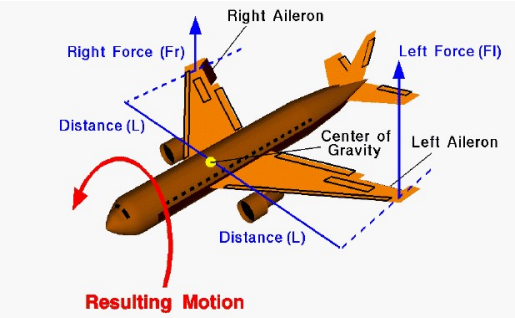
In the presence of wind, the velocity of the SUAV relative to the ground is:

$$\underline{v}_G^P = \underline{v}_W^P + \underline{v}_G^W$$
$$\begin{bmatrix} \dot{x}_{gr} \\ \dot{y}_{gr} \\ \dot{z}_{gr} \end{bmatrix} = \begin{bmatrix} \dot{x}_{air} \\ \dot{y}_{air} \\ \dot{z}_{air} \end{bmatrix} + \begin{bmatrix} \dot{x}_{wind} \\ \dot{y}_{wind} \\ \dot{z}_{wind} \end{bmatrix} \tag{12}$$

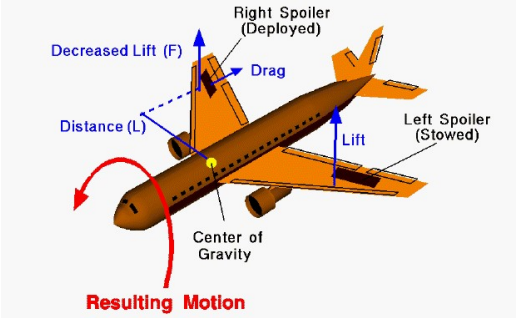
Therefore, prior to performing the integral in Eq. 10, all air velocity variables must be substituted using Eq. 11 to account for wind.

0.1.2 Flight direction control

The flight direction, or the yaw angle in this model, is altered by controlling the roll angle, ϕ of the SUAV, which is manipulated by the ailerons (Figure 0.6a) or spoilers (Figure 0.6b). This causes some of the lift to act as a centrifugal force as it takes a curved path.



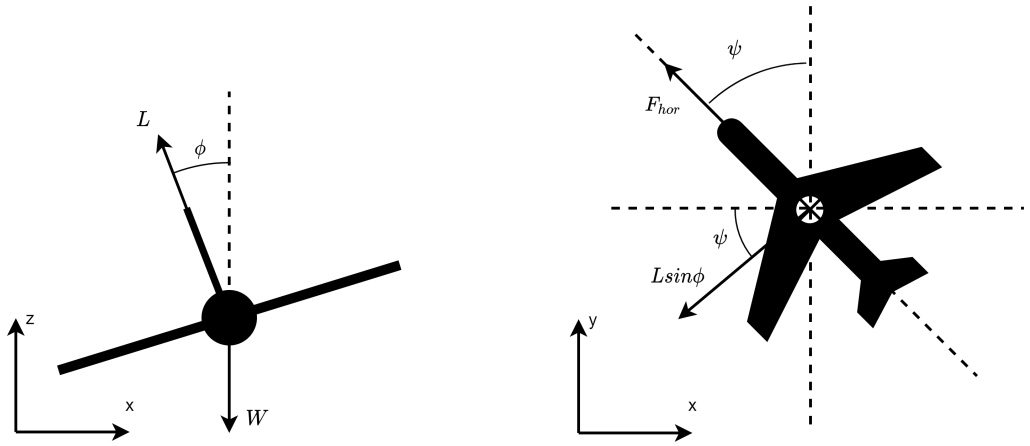
(a) Roll angle control via ailerons [3].



(b) Roll angle control via spoilers [4].

Figure 0.6: Methods for roll angle manipulation

The model assumes perfect control of the roll angle whereby ϕ is always equal to its demand. In reality, the roll angle is controlled by adjusting the torque generated from manipulating the ailerons and spoilers, causing angular acceleration in the direction of torque. This would be modelled by the implementation of another closed loop control system embedded within the flight model, but for simplicity, this has been negated. The effects of a roll angle is shown in Figure 0.7.



(a) Angle change in lift

(b) Additional lift force in horizontal plane

Figure 0.7: Effects on lift force due to a roll angle

In presence of a roll angle, the accelerations of the UAV relative to the ground become:

$$\ddot{x}_{gr} = -\frac{1}{m}[F_{hor}sin(\psi) - Lsin(\phi)cos(\psi)] \tag{13}$$

$$\ddot{y}_{gr} = \frac{1}{m}[F_{hor}cos(\psi) - Lsin(\phi)sin(\psi)] \tag{14}$$

$$\ddot{z}_{gr} = \frac{1}{m}[(T - D)sin(\gamma) + Lcos(\gamma)cos(\phi) - W] \tag{15}$$

As in Figure 0.7, ψ increases when ϕ is greater than 0 due to the horizontal lift component accelerating/decelerating the SUAV in the x and y directions. For instance, in the top left quadrant of Figure 0.7b, the horizontal lift increases $-\dot{x}_{gr}$ and decreases $\dot{y}_{gr}$, thereby increasing ψ. A simple closed loop control system (Figure 0.8) was implemented to control ψ through manipulation of ϕ.

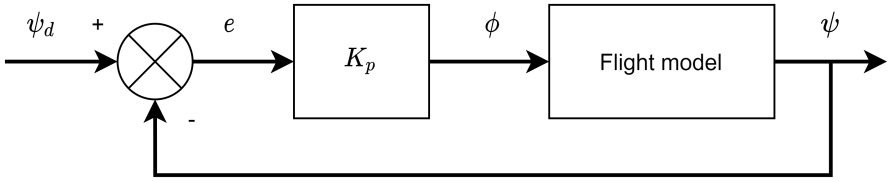


Figure 0.8: Closed loop control system for the SUAV yaw angle. ψ_d denotes the demand for ψ, and error, e, is the difference between the actual ψ and ψ_d.

This system implements a proportional controller with the proportional gain K_p. It assumes that ψ is perfectly measured, and that there is no delay between the actual ϕ and the required ϕ. With this system, and a large enough proportional gain, the overall path of the SUAV can be defined using a list of horizontal co-ordinates, $\underline{p}$, denoting the order of points to be visited by the SUAV, i.e.:

$$\underline{p} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

where n is the total number of points to visit, and x_n and y_n is the end point. Given the co-ordinate of the current horizontal location, x_t and y_t, and assuming the SUAV is heading to the first specified point, x_1 and y_1, ψ_d (Figure 0.9) can be determined.

$$\psi_d = tan^{-1}\left(\frac{x_1 - x_t}{y_1 - y_t}\right) \tag{16}$$

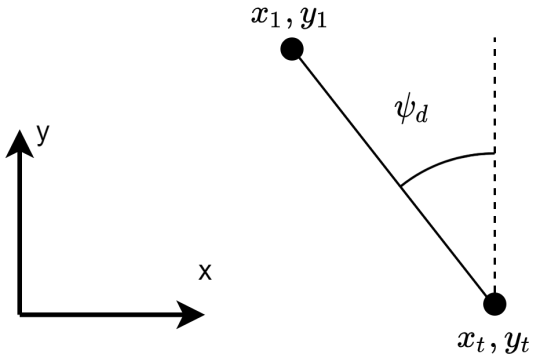


Figure 0.9: Illustration of ψ_d with respect to the SUAV current horizontal position and the target co-ordinate.

Upon reaching a user-defined radius, r_d, about the target co-ordinate, the SUAV will then move towards the next point in $\underline{p}$.

0.1.3 Power dynamics

The power inputs and outputs associated with the battery of the UAV are:

- Consumption for on-board electronics, P_{other}
- Consumption for propellers, P_{prop}
- Generation from solar energy, P_{solar}.

Therefore, the net power to the battery is:

$$P_{net} = P_{solar} - P_{prop} - P_{other} \tag{17}$$

By integrating the net power with respect to time, the energy level of the battery, E, at a point in time can be obtained:

$$E = E_0 + \int_{t_1}^{t_2} P_{net} dt \tag{18}$$

where E_0 is the initial battery energy. The solar power input is defined as:

$$P_{solar} = \eta_s I A_s \tag{19}$$

where η_s is the efficiency of the solar panels, I is the solar global horizontal irradiance (GHI), and A_s is the area of the solar panels. Specifically, GHI is the total amount of shortwave radiation experienced by a surface horizontal to the ground. In practice, the SUAV solar panels may not always be horizontal to the ground, as is the case when it performs any form of manoeuvring. However, this model assumes the duration at which the panels are perfectly horizontal is much greater than when the panels are skewed, as will be the case if the SUAV predominantly travels in a straight path. When the solar irradiance for clear skies, I_0, is known, the predicted irradiance under cloud cover, n can be estimated with the empirical formula [5]:

$$I = I_0(1 - 0.75n^{3.4}) \tag{20}$$

where n is a percentage value. Finally, the battery level of the SUAV may be referenced as state of charge:

$$SoC = \frac{E}{E_{max}} \times 100\% \tag{21}$$

where E_{max} is the battery capacity of the SUAV.

0.1.4 Flight Strategy

For a general point-to-point flight mission, two flight strategies were employed:

1. Constant Altitude Flight (CAF)

   - The SUAV ascends with constant vertical velocity by propelling itself with a multiplier (λ_{asc}) of the cruise power, P_{cruise}.
   - Upon reaching a defined cruise altitude, z_{cruise}, the SUAV maintains altitude until the final point is reached, propelling at P_{cruise}.
   - It begins to steadily descend right after reaching the final point with a multiplier (λ_{dsc}) of P_{cruise}, circling the final point as it descends.

2. Variable Altitude Flight (VAF)

   - The SUAV ascends with a multiplier (λ_{asc}) of P_{cruise}, until z_{cruise} is reached, where it will propel at P_{cruise}.
   - If P_{net} is positive and the battery is fully charged, the surplus power is fed to the propeller, thus raising the altitude of the SUAV until a defined altitude limit, $z_{cruise,max}$, and storing the excess energy in the form of gravitational potential energy. Upon reaching $z_{cruise,max}$, the SUAV will cruise.
   - If P_{net} is negative or the battery is not fully charged, the SUAV will gradually descend and consume less propelling power than P_{cruise} while it is still flying above z_{cruise}. Otherwise, it will cruise at z_{cruise}.
   - It begins to steadily descend right after reaching the final point with a multiplier (λ_{dsc}) of P_{cruise}, circling the final point as it descends.

P_{cruise} is defined as (derived in Appendix A):

$$P_{cruise} = \frac{1}{\eta_{prop}} \frac{C_D}{C_L^{\frac{3}{2}}} \sqrt{\frac{2m^3 g^3}{\rho A}} \tag{22}$$

where applying this amount of power to the propellers will cause the SUAV to fly at a fixed altitude and speed. In hindsight, the VAF strategy borrows the concept of storing excess power in the form of altitude from Lee and Yu's [6] planner, while the idea behind the CAF strategy is simplicity. The flight strategies are illustrated as process flows in Figures 0.10 and 0.11.

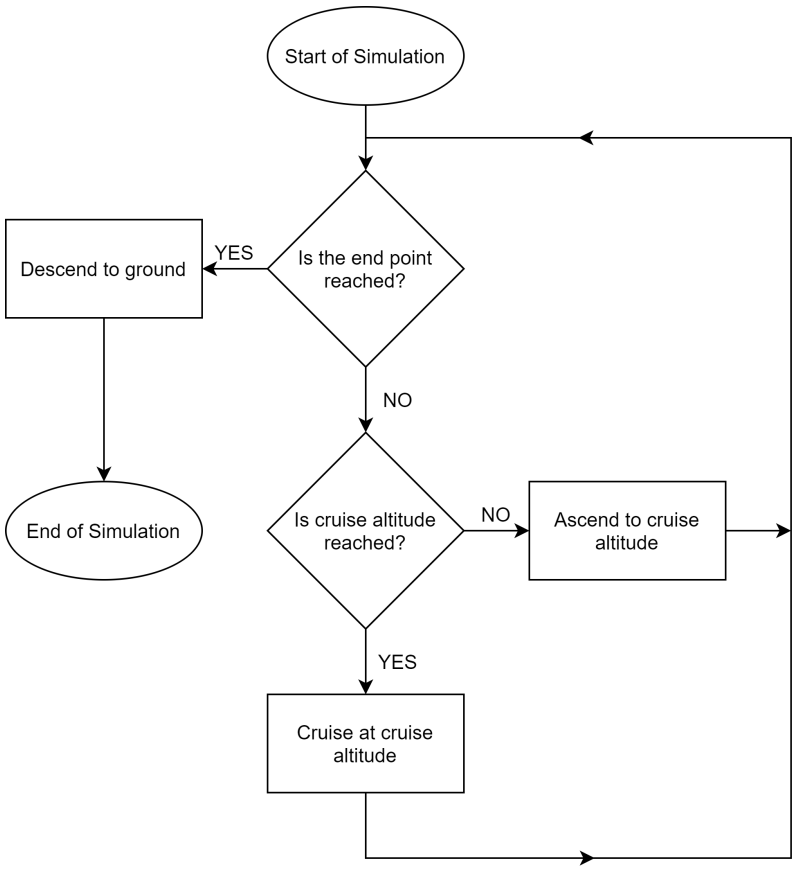


Figure 0.10: Constant Altitude Flight simulation process flow.

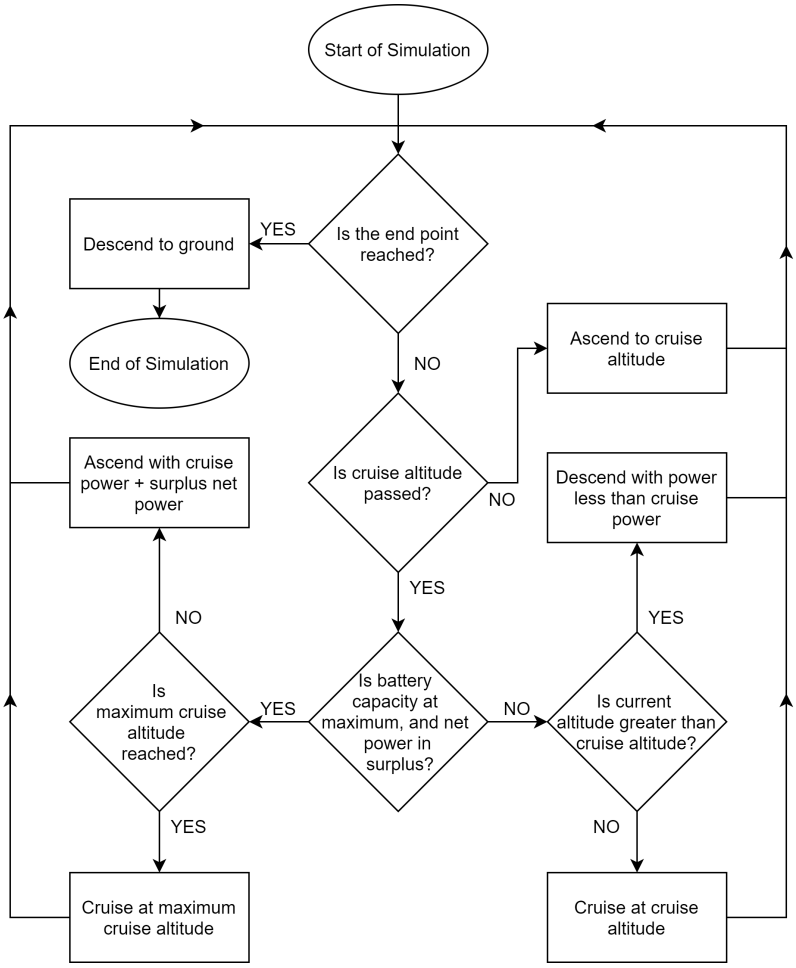


Figure 0.11: Variable Altitude Flight simulation process flow.

Predominantly, the SUAV would most likely begin descending as it approaches its landing/end point in an actual flight. This model assumes a pessimistic flight descent pattern, and would generally consume more power and time to descend, compared to a direct descent. At any point of the simulation, the propelling power is zero whenever the battery level is flat. Furthermore, the decreasing air density with increasing altitude is accounted for in this flight model using the values specified in Appendix B.

0.2 Numerical Modelling for Flight Simulation

Fundamentally, the problem the flight model tackles can be represented in a single question: given particular starting flight conditions and mission parameters, how will the SUAV position, velocity, and battery energy vary as the simulation proceeds? These variables can be solved by integration:

$$\underline{d}_G^P = \int_{t_1}^{t_2} \underline{v}_G^P dt = \begin{bmatrix} x_{gr} \\ y_{gr} \\ z_{gr} \end{bmatrix} \tag{11 revisited}$$

$$\underline{v}_G^P = \int_{t_1}^{t_2} \underline{\dot{v}}_G^P dt = \begin{bmatrix} \dot{x}_{gr} \\ \dot{y}_{gr} \\ \dot{z}_{gr} \end{bmatrix} \tag{10 revisited}$$

$$E = E_0 + \int_{t_1}^{t_2} P_{net} dt \tag{18 revisited}$$

To simulate flight and solve for the SUAV position, velocity and battery energy with respect to time, time-stepping integration was performed on the above equations using an Ordinary Differential Equation (ODE) solver. The well-established Runge-Kutta 45 (RK45) method was chosen as the solver for the following due to its fourth-order accuracy; since the flight model equations is second-order at the highest, numerical error can be minimised to ensure stability when performing simulations [7].

The flight model system of ODEs were decoupled to implement the RK45 method, whereby the state variables, $\underline{s}_v$, and state derivatives $\underline{s}_v$ are defined as:

$$\underline{s}_v = \begin{bmatrix} x_{gr} \\ y_{gr} \\ z_{gr} \\ \dot{x}_{gr} \\ \dot{y}_{gr} \\ \dot{z}_{gr} \\ E \end{bmatrix}, \quad \underline{s}_d = \begin{bmatrix} \dot{x}_{gr} \\ \dot{y}_{gr} \\ \dot{z}_{gr} \\ \ddot{x}_{gr} \\ \ddot{y}_{gr} \\ \ddot{z}_{gr} \\ P_{net} \end{bmatrix} \tag{23}$$

where $\ddot{x}_{gr}$, $\ddot{y}_{gr}$, $\ddot{z}_{gr}$, and P_{net} are as previously defined in Eq.s 3.13, 3.14, 3.15, and 3.17 respectively.

0.3 Monte Carlo Simulations

There are near countless combinations of cloud cover and wind when accounting for uncertainty. Consider a hypothetical flight with 4 segments containing possible values for cloud cover (CC) and wind velocity within each segment in Figure 0.12. Each flight segment has a predicted cloud cover and wind velocity value (indicated by bold numbers), while an arbitrary level of uncertainty is indicated by other potential values.

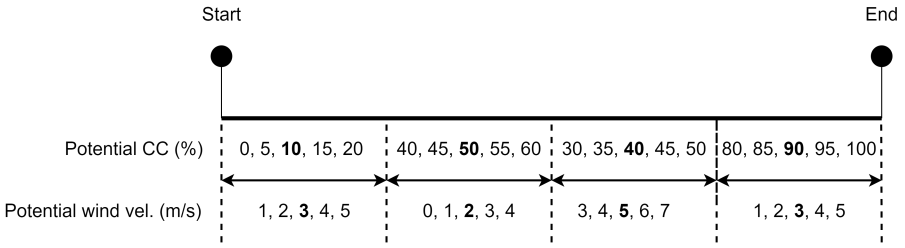


Figure 0.12: Representation of possible wind and cloud cover values for a hypothetical point-to-point flight.

The total number of combinations of cloud cover and wind velocity for this example is 5^8 ($= 390625$), all of which are impossible to simulate within a reasonable time frame. Assuming an optimistic simulation time of one second for each combination, it will require 108 hours to simulate flights with every combination. Therefore, the Monte Carlo method was employed to the simulations, such that an approximate solution to the success rate of a mission can be achieved.

The Monte Carlo method relies on repeated sampling to evaluate numerical results, whereby the underlying principle is to utilise randomness to generate samples and results that are representative and approximate [8]. In the context of simulating flights represented by Figure 0.12, the Monte Carlo simulations conducted to determine the flight mission success rate will take the following procedure:

1. Sample cloud cover and wind velocity combination.
2. Run simulation with sampled combination, and determine if flight is successful or not.
3. Repeat 1. and 2. until simulations are repeated for a specified number of times.
4. Calculate mission success rate: $\frac{no.\ of successful\ flights}{no.\ of\ simulations}$

Generally, the total number of simulations are set as high as possible to generate more accurate answers.

0.3.1 Wind Data Sampling

Hourly horizontal wind data was sampled and recreated by finding the mean and standard deviation of the hourly difference in historic wind velocity (μ_{WV}, σ_{WV}) and direction (μ_{WD}, σ_{WD}) data, along with the maximum wind velocity, $v_{W,max}$. This will define the normal probability distribution for hourly difference in velocity and direction. The hourly difference is the difference between the velocity (or direction) of the current hour and the previous hour - consider Table 0.1 which indicates how the hourly difference from an example wind data is evaluated. Moreover, the wind direction in the context of this model is the clockwise angle from the positive y-direction, similar to the yaw angle.

Table 0.1: Example hourly data.

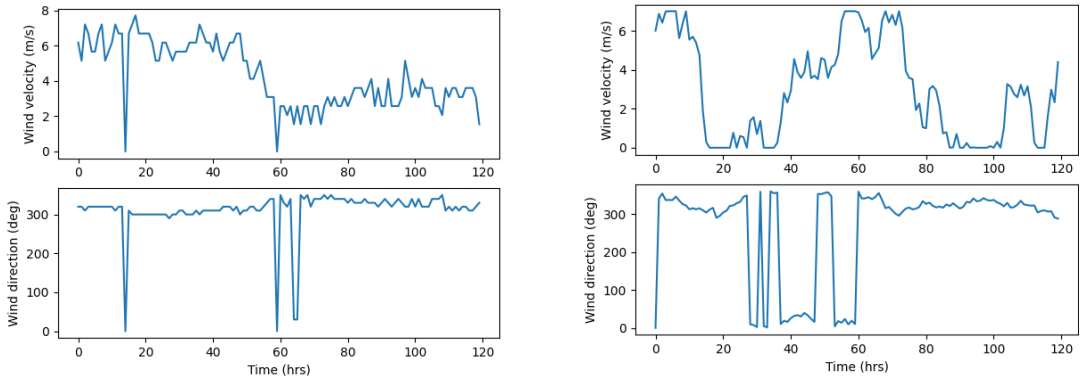
Hour	Wind Velocity (m/s)	Hourly Diff. (Vel.)	Wind Direction(°)	Hourly Diff. (Dir.)
1	5	N/A	50	N/A
2	7	+2	55	+5
3	2	-5	40	-15
4	6	+4	50	+10
5	4	-2	45	-5

Hence, the wind data sampling procedure is:

1. Determine μ_{WV}, σ_{WV}, μ_{WD}, σ_{WD}, and $v_{W,max}$ from historic data.
2. Specify the initial wind velocity, $v_{W,0}$, wind direction, $\psi_{W,0}$, starting hour, and duration for the sample wind data.
3. Draw random samples of the hourly difference in velocity and direction from their respective probability distributions in each hour excluding the starting hour.
4. Build up the wind data using the hourly difference values, starting from the initial wind velocity and direction. If the calculated wind velocity exceeds $v_{W,max}$, the wind velocity is set to $v_{W,max}$. Additionally, if it is less than zero, the wind velocity is set to zero.

For instance, if at hour 1, the initial wind velocity is 3, and the sampled hourly difference for velocity in hour 2 is +3, then the wind velocity in hour 2 is evaluated to be 6. This process is then repeated until the final hour in the data. Figure 0.13 shows a comparison of historic hourly wind data in Aberdeenshire, UK with a generated sample of wind data using μ_{WV}, σ_{WV}, μ_{WD}, σ_{WD}, and $v_{W,max}$ derived from the historic wind data.

With this sampling method, uncertainty in the wind conditions for a given flight mission is represented with the variation among different wind data samples.



(a) Historic wind data.

(b) Generated sample wind data.

Figure 0.13: Comparison of historic and generated wind data.

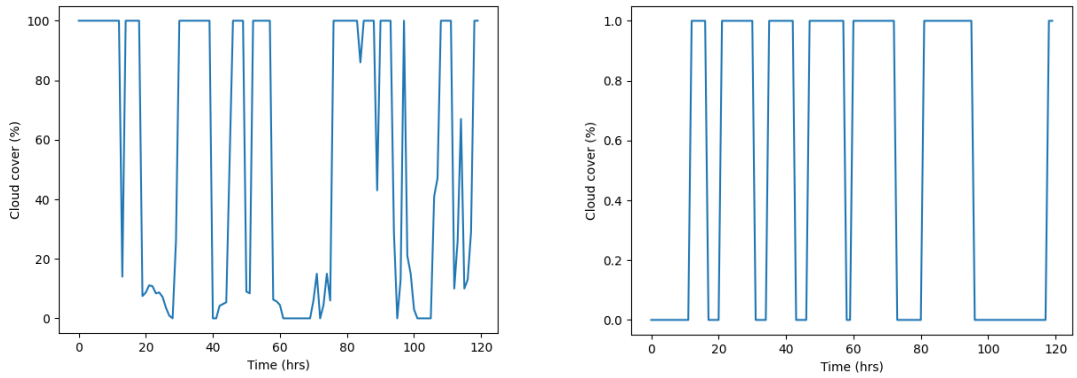
0.3.2 Cloud Cover Data Sampling

Hourly cloud cover data was sampled and recreated by finding the mean and standard deviation of the periods at which the cloud cover is 100% ($\mu_{CC,100}$, $\sigma_{CC,100}$) and not 100% ($\mu_{CC,0}$, $\sigma_{CC,0}$) in historic cloud cover data. To simplify the sampling process, the sampled cloud cover data is only either 0% or 100%, and thus, when evaluating the means and standard deviations, any cloud cover percentage lower than 100% is considered as 0%. The cloud cover data sampling procedure is:

1. Determine $\mu_{CC,100}$, $\sigma_{CC,100}$, $\mu_{CC,0}$, and $\sigma_{CC,0}$ from historic data,
2. Specify the initial cloud cover (0% or 100%), starting hour, and duration for the sample cloud cover data.
3. If initial cloud cover is 0%, draw random sample of period at which cloud cover is 0% from the normal distribution defined by $\mu_{CC,0}$ and $\sigma_{CC,0}$. Extend the 0% cloud cover by the sampled period. If initial cloud cover is 100%, this step is the same, except a random sample is drawn from the normal distribution defined by $\mu_{CC,100}$ and $\sigma_{CC,100}$.
4. Alternate to the next cloud cover percentage, and draw from the respective normal distribution to build up the cloud cover data until the specified duration is met.

Figure 0.14 compares historic cloud cover data over London with a generated sample of cloud cover data using the means and distributions derived from the historic data.

Uncertainty in the cloud cover for a particular flight mission is implemented with the variation among cloud cover data samples.



(a) Historic cloud cover data.

(b) Generated sample cloud cover data.

Figure 0.14: Comparison of historic and generated cloud cover data.

0.4 Software Implementation

The flight model was developed in the *Python* programming language. At top-level, the tasks performed by the flight simulation model are straightforward, and are shown in Figure 0.15 below.

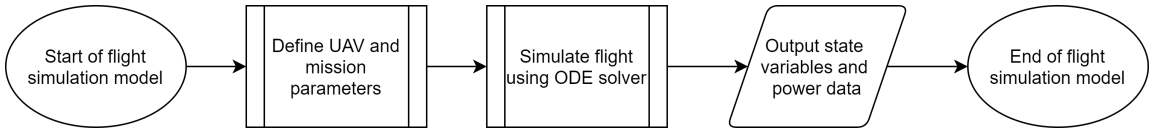


Figure 0.15: Top-level processes carried out by the flight simulation model.

The 'Define UAV and mission parameters' subroutine primarily involves determining the variables required to evaluate the flight equations within the ODE solver, such as mass, wing area, and flight points, amongst other variables. An exhaustive list of the variables defined and the method used to define certain variables is illustrated in Figure 0.16.

The hourly solar irradiance data retrieved with the *pvlib* package is based on the first date of the specified month, while the location is defined by latitudinal and longitudinal co-ordinates. Using the defined parameters, flight is simulated through use of the ODE solver under the *scipy* package, which performs a sequence of calculations at each time-step to determine the state variables at the next time step until the SUAV lands ($z_{gr} = 0$). This 'Simulate flight using ODE solver' subroutine is elaborated in Figure 0.17.

The output state variables and power data can then be used in conjunction with the *matplotlib* package to illustrate the flight simulation results as plots.

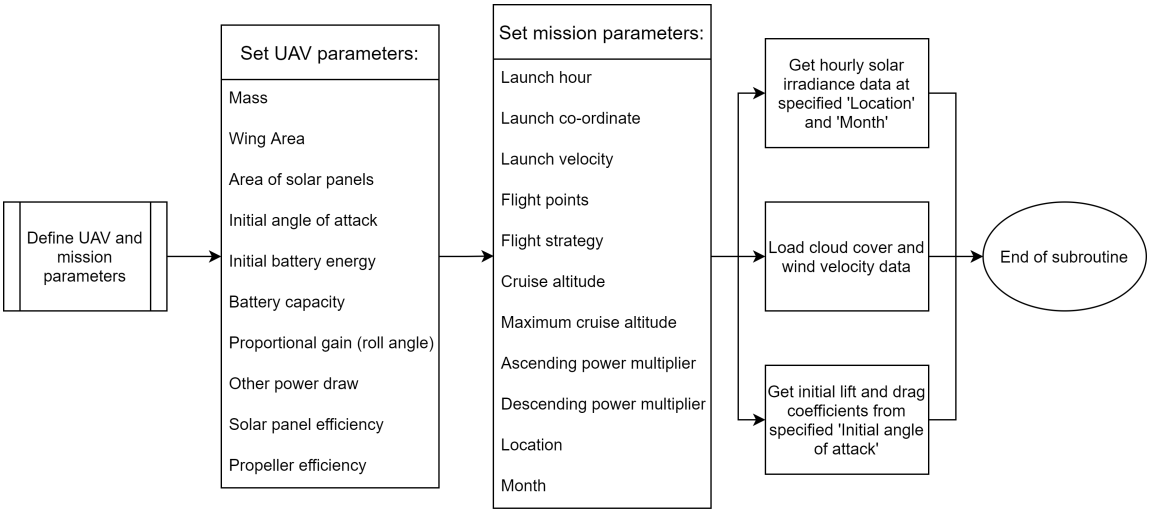


Figure 0.16: 'Define UAV and mission parameters' subroutine.

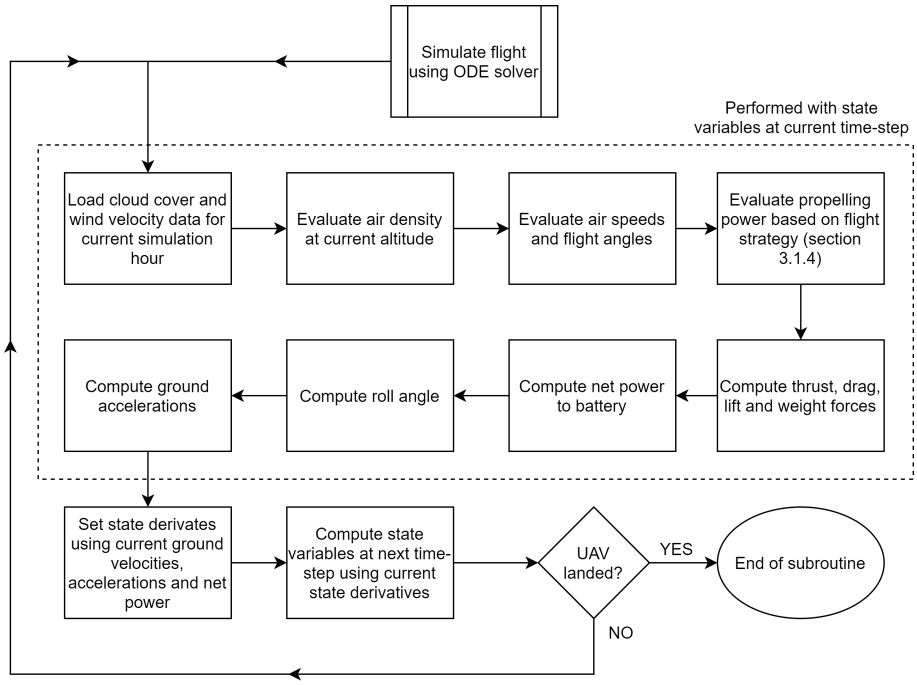


Figure 0.17: 'Simulate flight using ODE solver' subroutine.

Bibliography

[1] *Aircraft Principal Axes*. URL: https://commons.wikimedia.org/wiki/File:Yaw_Axis_Corrected.svg.

[2] Tom Benson. *Rudder - Yaw*. URL: https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/rud.html.

[3] Tom Benson. *Ailerons*. URL: https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/alr.html.

[4] Tom Benson. *Spoilers*. URL: https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/spoil.html.

[5] Air Pollution Training Institute. *Solar Radiation Cloud Cover Adjustment Calculator*. 2004. URL: http://www.shodor.org/os411/courses/_master/tools/calculators/solarrad/.

[6] Joo-Seok Lee and Kee-Ho Yu. "Optimal Path Planning of Solar-Powered UAV Using Gravitational Potential Energy". In: (Feb. 2017). DOI: 10.1109/TAES.2017.2671522.

[7] J.R. Dormand and P.J. Prince. "A family of embedded Runge-Kutta formulae". In: *Journal of Computational and Applied Mathematics* 6.1 (1980), pp. 19–26. ISSN: 0377-0427.

[8] A.M. Johansen. "Monte Carlo Methods". In: *International Encyclopedia of Education (Third Edition)*. Ed. by Penelope Peterson, Eva Baker, and Barry McGaw. Third Edition. Oxford: Elsevier, 2010, pp. 296–303. ISBN: 978-0-08-044894-7.