Assignment 3 Presentation

Ravi Kumar — Vighnesh J R — Shreyas N B — Mayank Ghritlahre

November 3, 2024

AE667 Rotary Wing Aerodynamics - Group 1

Team Contribution

Roll Number	Name	Contribution	Remarks
210010052	Ravi Kumar	5	Code
210010073	Vighnesh J R	5	Theory
210010061	Shreyas N B	5	Code, Slides
210010040	Mayank Ghritlahre	5	Slides

Assumptions and Data

Physics Assumptions/Data

- The flight simulator updates parameters, forces, and moments every second.
- Gravity remains constant with altitude ($g = 9.81\ m/s^2\ \widehat{Z}_{hel}$).

Environmental Assumptions/Data

- We apply the ISA (International Standard Atmosphere) model, linearly interpolating values based on helicopter altitude.

Flight Condition Assumptions/Data

- The helicopter is in level flight at a constant altitude with no wind.

Vehicle and Flight Condition Assumptions

Vehicle Assumptions/Data

- Inviscid, compressible $\left(\frac{1}{\sqrt{1-M_\infty^2}}\right)$ flow is assumed for the rotors.
- The engine is Turbotech, using Jet A1 fuel with a calorific value of 43.124 MJ/kg (based on a given data model).
- NACA0012 C_L vs α $(0 - 360°)$ data was obatined using WebPlotDigitizer from a given graph.
- Mass of vehicle is 50 kg fuel, 50 kg payload and 100 kg structural weight.

Algorithm/Logic Flow Diagrams: mission.py

mission.py

class Mission

private

__collective_pitch() # Objective fxn in θ_0 calculation
__lambda_i() # Glauert induced velocity
__get_cp_from_force # Collective pitch for given thrust (forward)

details

simulate() ← input ← mission.toml # has mission segments
params.toml # has helicopter params

runs

mission segment computers

takeoff()
vertical_climb_descent_and_hover() # BEMT
steady_climb_descent_and_hover() # forward

computes θ_0 from $T = W$ and $\theta_{0,t}$ from trim

↓

power consumption

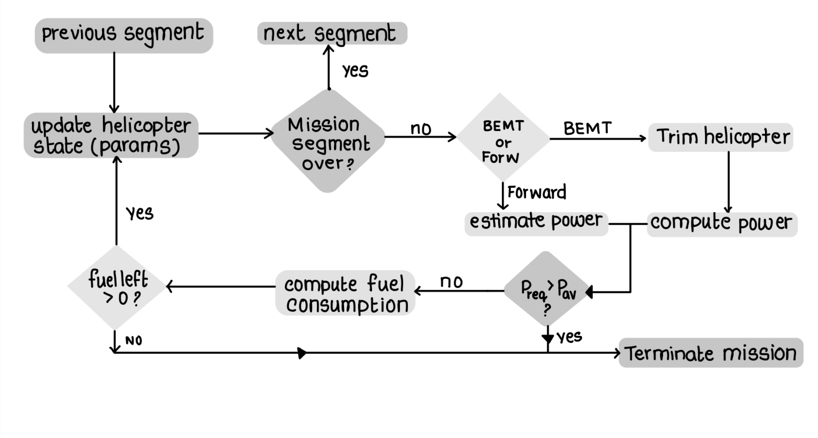
change_payload() # increments/drops total mass

there are functions to plot data

Algorithm/Logic Flow Diagrams: mission.py

1. The mission profile is provided through `mission_A.toml`,`mission_B.toml`, etc., files which contains the rates and distances for each individual segment.
2. The script accounts for switching between forward flight (Glauert model) and hover (BEMT model) based on the mission profile, and also uses a dynamic time scale.
3. The script calculates the gross weight, fuel consumption, fuel burn rate, altitude, power, speed, climb rate, and distance covered at each time interval.

Algorithm/Logic Flow Diagrams: Mission computer



Algorithm/Logic Flow Diagrams: helicopter.py

This file combines dynamics from the two rotors, two stabilizers and the fuselage. It is initialized with a `params.toml` file as shown, and creates 2 `RotorBlade` objects for the main and tail rotors, and 2 `Wing` objects for the horizontal and vertical stabilizers.

Control inputs are fed into these objects, and we use their functions described earlier to calculate the forces, moments, power etc.

Almost all the quantities are in a vector form, so we just use vector algebra to sum them up and get the net quantities, which are then printed in a neat format to ease manual trimming (done through running `python helicopter.py`).

Algorithm/Logic Flow Diagrams: helicopter.py

The `Wing` object is used to initialize the horizontal and vertical stabilizers using the design parameters given. We calculate the net lift and drag by using the net velocity over each and add it to the net force and moment.

This is done by accounting for the wake of the main rotor flowing over the stabilizers, and we obtain this wake depending on the angle of attack at each radially discrete point on the TPP. If the wake is not able to reach the stabilizer at any point, then V_∞ is used instead.

A visualization of this is shown in the next slide.

Algorithm/Logic Flow Diagrams: helicopter.py

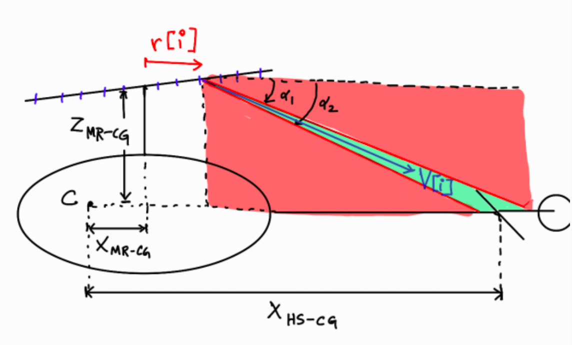


Figure 1: Main rotor wake on stabilizers

Hence, we only consider the wake of such elements which flow over the stabilizers (green region) by accounting for the angle of attack at each point on the TPP, i.e., $\alpha_1 \leq \alpha[i] \leq \alpha_2$.

Algorithm/Logic Flow Diagrams: trim_curve.ipynb

Here we use the simplified power coefficient expression to estimate the main and tail rotor power at each time interval in various flight conditions.

$$C_p = \kappa C_T \lambda_i + \frac{1}{2}\frac{f}{A}\mu^3 + \frac{\sigma C_{d0}}{8}[1 + 4.6\mu^2]$$

During the mission planner test, we will have access to only approximate solutions for C_T, μ, λ_i and α_{TPP}. So, we trim the helicopter **but** use the below approximate expressions to fit the power coefficient curve.

$$C_T = \frac{\sqrt{W^2 + D^2}}{\rho_\infty (\Omega R)^2} \quad \alpha_{\mathrm{TPP}} = \tan^{-1}\frac{D}{W} \quad \mu = \frac{V_\infty \cos\alpha_{\mathrm{TPP}}}{\Omega R}$$

Algorithm/Logic Flow Diagrams: trim_curve.ipynb

λ_i needs to be solved iteratively and is dependent on $\mu, C_T, \alpha_{\mathrm{TPP}}$. We do this by defining an objective function `__lambda_i` which takes $\mu, C_T, \alpha_{\mathrm{TPP}}$ and uses `fsolve` to solve for λ_i inside the `cal_median_forward_flight_params` function.

Also, we fit only the main rotor power curve and then calculate the ratio between the main and tail rotor power and fit a straight line to it to get the tail rotor power as a function of main rotor power. This was done as we are not fully convinced of fitting the curve for tail rotor after neglecting the fuselage drag term.

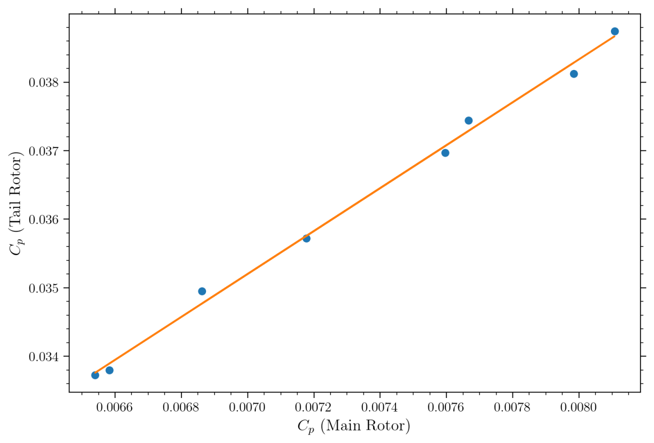


Figure 2: Tail rotor power vs Main rotor power

Algorithm/Logic Flow Diagrams: trim_curve.ipynb

V_inf(kmph)	Altitude(m)	weight(kg)	True Values				Low Fidelity Values				solidity(MR)	solidity(TR)	Cp(MR)	Cp(TR)	Cp(Total)
			Ct	Lambda_i	Cp_parasite	mu	Ct	Lambda_i	Cp_parasite	mu					
50	2000	200	1.59E-03	1.03E-01	1.75E-06	5.55E-02	1.59E-03	1.38E-02	1.75E-06	5.55E-02	5.73E-02	6.37E-02	7.60E-03	3.70E-02	7.78E-03
50	2000	150	1.20E-03	8.38E-02	1.75E-06	5.55E-02	1.19E-03	1.05E-02	1.74E-06	5.55E-02	5.73E-02	6.37E-02	7.18E-03	3.57E-02	7.36E-03
50	2500	200	1.68E-03	1.07E-01	1.75E-06	5.55E-02	1.67E-03	1.45E-02	1.75E-06	5.55E-02	5.73E-02	6.37E-02	7.67E-03	3.74E-02	7.86E-03
50	2500	150	1.26E-03	8.56E-02	1.75E-06	5.55E-02	1.25E-03	1.10E-02	1.74E-06	5.55E-02	5.73E-02	6.37E-02	6.86E-03	3.50E-02	7.04E-03
40	2000	200	1.59E-03	1.22E-01	8.94E-07	4.44E-02	1.59E-03	1.67E-02	8.94E-07	4.44E-02	5.73E-02	6.37E-02	8.11E-03	3.87E-02	8.31E-03
40	2000	150	1.20E-03	9.53E-02	8.94E-07	4.44E-02	1.19E-03	1.28E-02	8.94E-07	4.44E-02	5.73E-02	6.37E-02	6.54E-03	3.37E-02	6.71E-03
40	2500	200	1.67E-03	1.25E-01	8.94E-07	4.44E-02	1.67E-03	1.74E-02	8.94E-07	4.44E-02	5.73E-02	6.37E-02	7.98E-03	3.81E-02	8.18E-03
40	2500	150	1.26E-03	9.76E-02	8.94E-07	4.44E-02	1.25E-03	1.34E-02	8.94E-07	4.44E-02	5.73E-02	6.37E-02	6.58E-03	3.38E-02	6.76E-03

Figure 3: Table of altitudes, gross weights and trim settings

The complete values can be found at `data/trim_curve.csv`. We get the following values after fitting:

$$\kappa \to 25.85 \quad C_{d0} \to 0.48 \quad 4.6 \to 139.52$$

$$C_{P_{\text{tail rotor}}} = 3.13 C_{P_{\text{main rotor}}} + 1.33 \times 10^{-2}$$

Algorithm/Logic Flow Diagrams: optimize_mission.py

This code simply runs the individual mission profile A in a loop with varying maximum altitudes in order to determine the maximum altitude at which the mission can be completed. This is done to ease some load on manually finding this altitude.

Algorithm/Logic Flow Diagrams

Note: Please refer to the code files for further understanding and the functions' inputs/outputs, as there are various internal functions not shown in the diagrams. Each function has a docstring explaining its inputs and purpose. The variables are also named in a self-explanatory manner.

Test Vehicle

The design parameters have been chosen same as in Assignment-2.

$$\alpha_{\text{fuselage}} = 1^\circ$$

Parameter	Main Rotor	Tail Rotor
Airfoil	NACA 0012	NACA 0012
Rotor radius (m)	2.5	0.5
Rotor speed (rad/s)	100	400
Number of blades	3	2
Root cutout (m)	0.1	0.025
Chord length variation	$0.2 \rightarrow 0.10$	$0.05 \rightarrow 0.05$
Twist variation (m^{-1})	0°	0°

Test Vehicle

Parameter	Horizontal Stabilizer	Vertical Stabilizer
Airfoil	NACA 0012	NACA 0012
Span (m)	0.8	0.7
Chord (m)	0.2	0.3

Test Vehicle

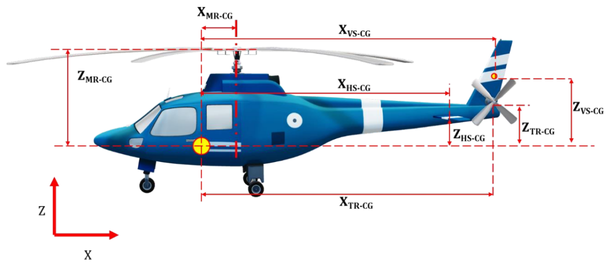
The distances of the stabilizers have been chosen differently from given parameters, so that it matches our Assignment 2 helicopter.

$$X_{MR-CG} = 0\text{m};\ X_{TR-CG} = 4.75\text{m}$$

$$Z_{MR-CG} = 1.5\text{m};\ Z_{TR-CG} = 1\text{m}$$

$$X_{HS-CG} = 3.5\text{m};\ X_{VS-CG} = 4.75\text{m}$$

$$Z_{HS-CG} = 1\text{m};\ Z_{VS-CG} = 1\text{m}$$



Trim

For a 50 km/h level flight at 2000 m AMSL with $\alpha_{\text{fuselage}} = 1^\circ$ on addition of stabilizers:

	(degrees)		(N or N-m, Vehicle)
$\theta_{o,m}$	2.267	F_X	-0.008
θ_{1c}	1.55	F_Y	25.05
θ_{1s}	2.62	F_Z	0.56
$\theta_{o,t}$	2.6	M_X	12.46
α_{TPP}	1.167	M_Y	16.01
β_o	0.3	M_Z	-0.78

Table 1: Trim Settings

Mission planner test

Mission A: Successful payload drop mission

We assume the following rates and distances for this mission:

Vertical climb	Climb velocity = 1m/s
Steady climb	Climb velocity = 4 m/s Forward Velocity = 9 m/s
Level Flight	Forward Velocity = 14m/s Distance = 5 km
Steady descent	Climb velocity = -0.5m/s Forward Velocity = 9 m/s
Vertical descent	Climb velocity = -0.5m/s

Mission planner test: Mission A plots

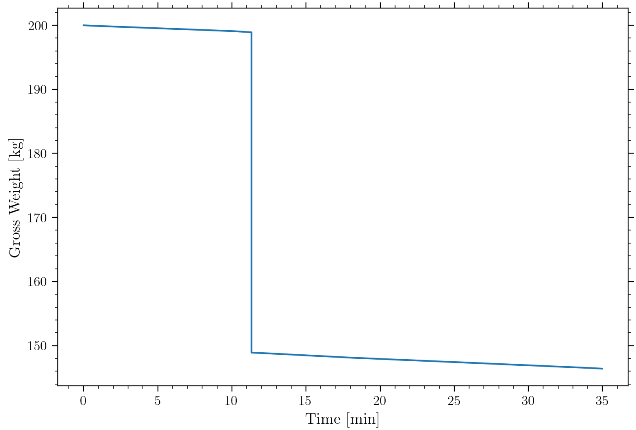


Figure 5: Gross Weight vs Time

Mission planner test: Mission A plots

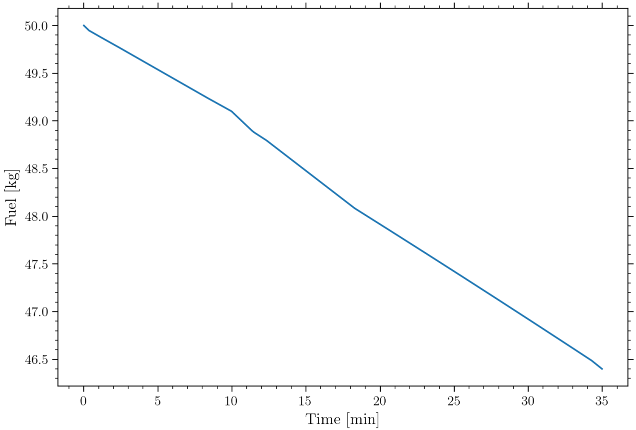


Figure 6: Fuel vs Time

Mission planner test: Mission A plots

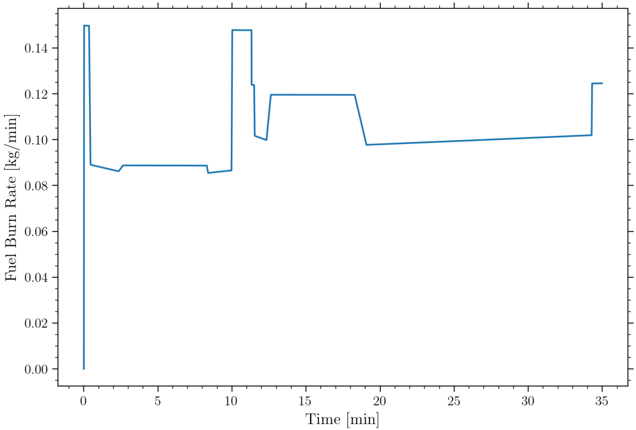


Figure 7: Fuel Burn Rate vs Time

Mission planner test: Mission A plots

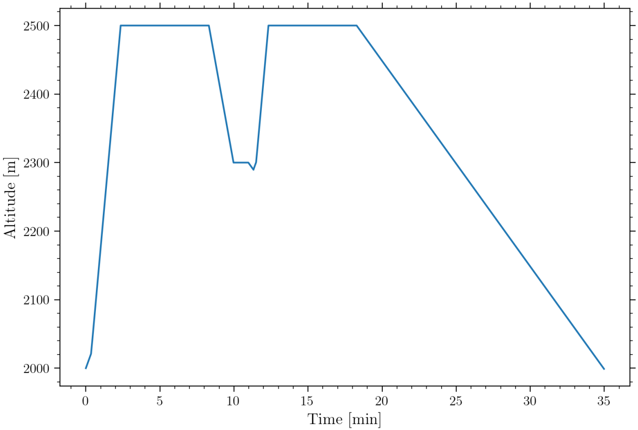


Figure 8: Altitude vs Time

Mission planner test: Mission A plots

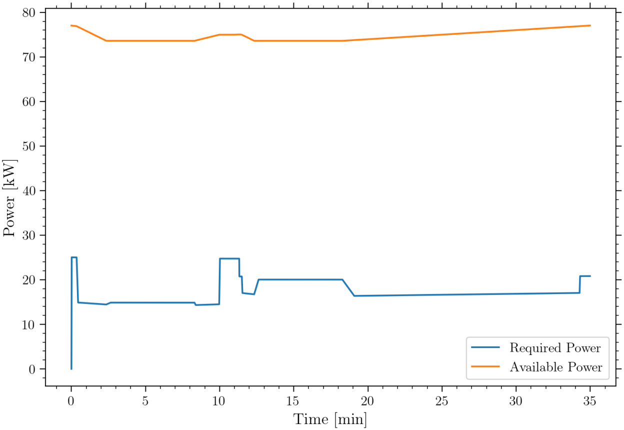


Figure 9: Power vs Time

Mission planner test: Mission A plots

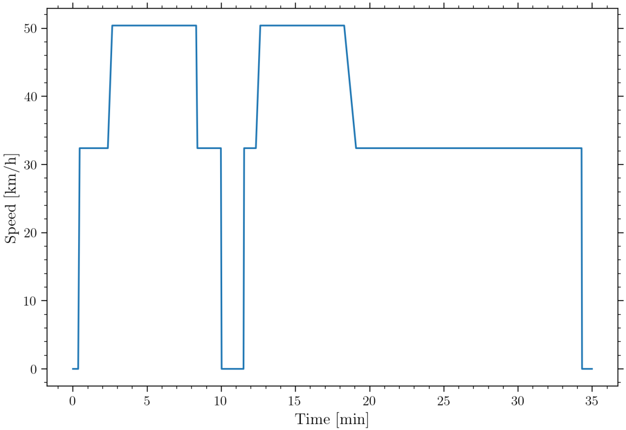


Figure 10: Speed vs Time

Mission planner test: Mission A plots

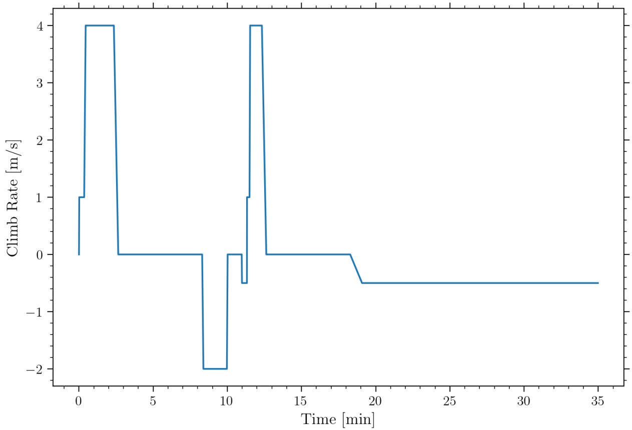


Figure 11: Climb Rate vs Time

Mission planner test: Mission A plots

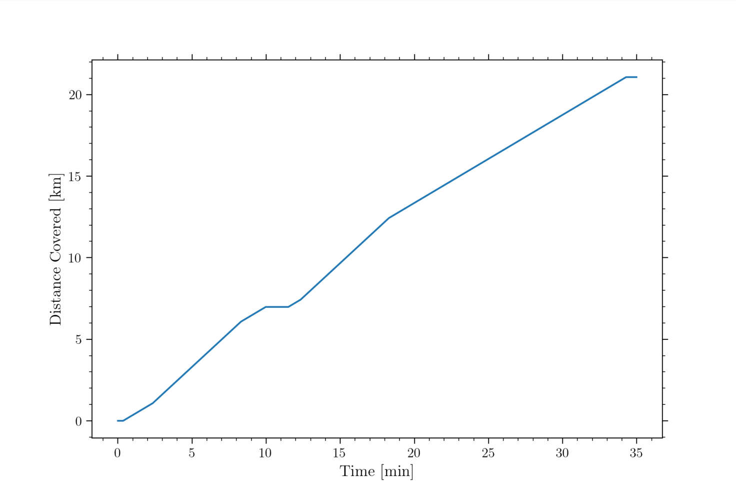


Figure 12: Distance Covered vs Time

Mission B: Successful payload pickup mission

We assume all the same rates and distances as in Mission A, except the fact that takeoff payload is 0 kg and payload is picked up instead of being dropped.

Mission planner test: Mission B plots

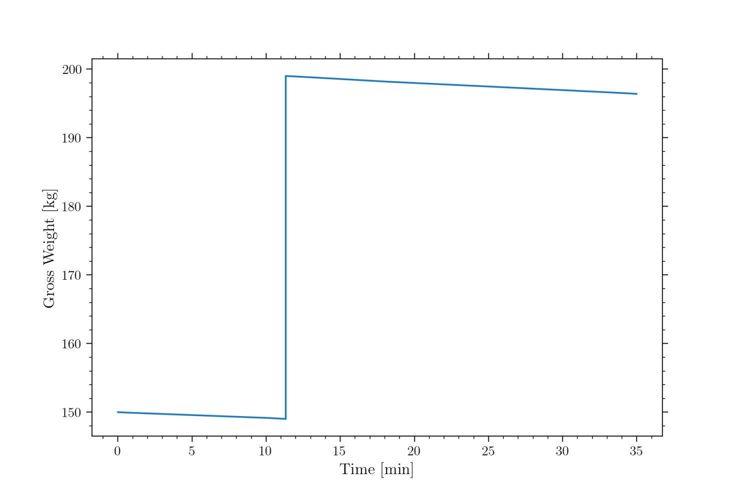


Figure 13: Gross Weight vs Time

Mission planner test: Mission B plots

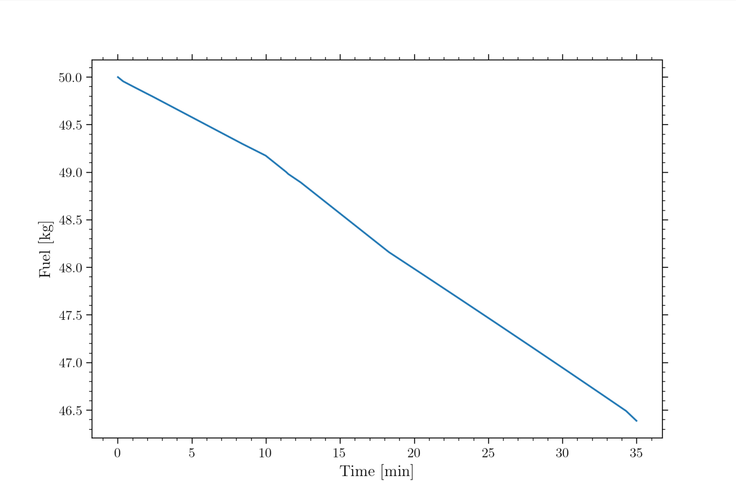


Figure 14: Fuel vs Time

Mission planner test: Mission B plots

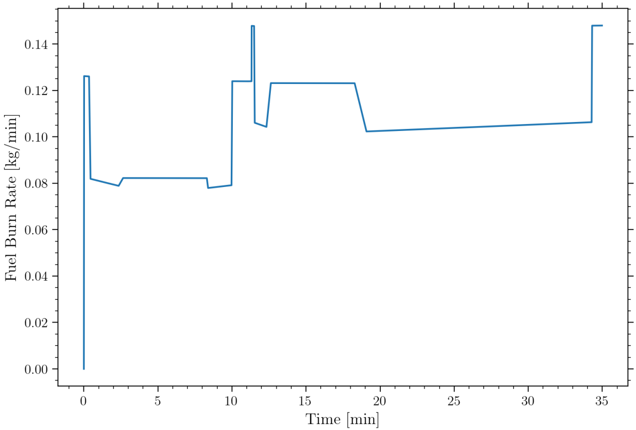


Figure 15: Fuel Burn Rate vs Time

Mission planner test: Mission B plots

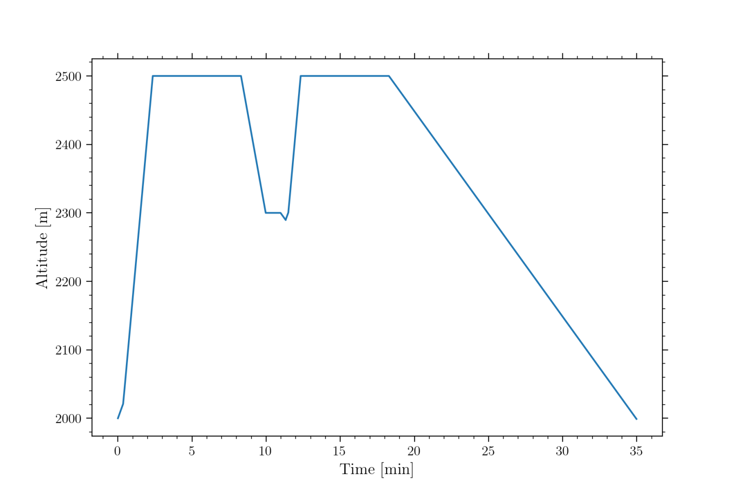


Figure 16: Altitude vs Time

Mission planner test: Mission B plots

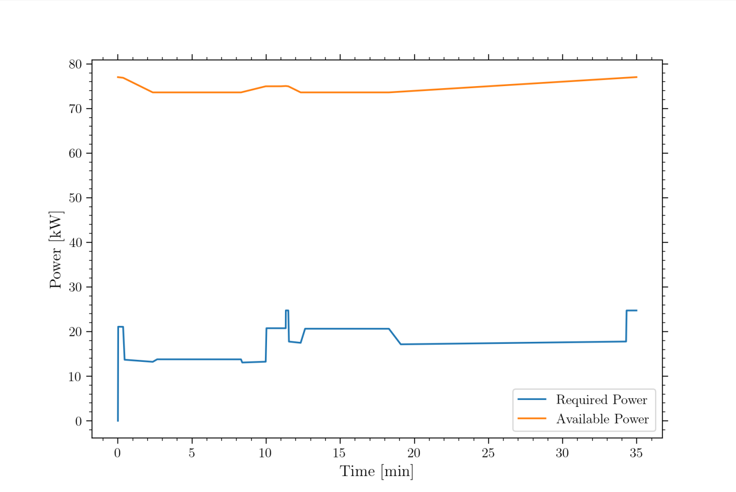


Figure 17: Power vs Time

Mission planner test: Mission B plots

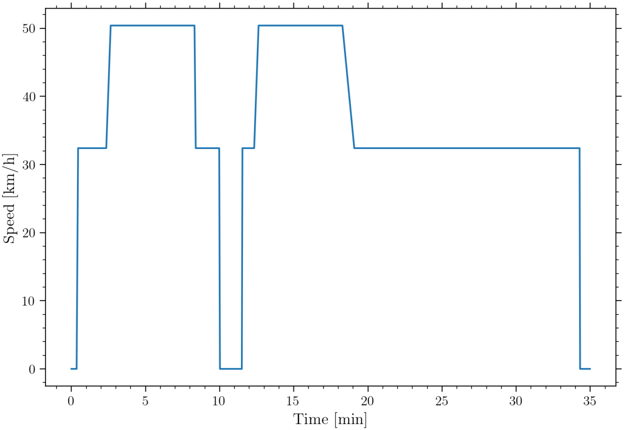


Figure 18: Speed vs Time

Mission planner test: Mission B plots

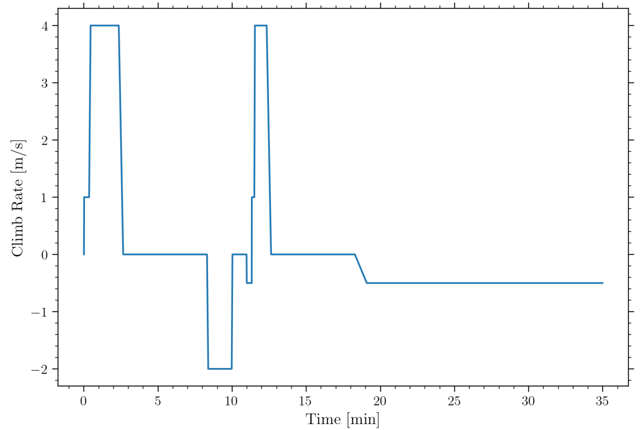


Figure 19: Climb Rate vs Time

Mission planner test: Mission B plots

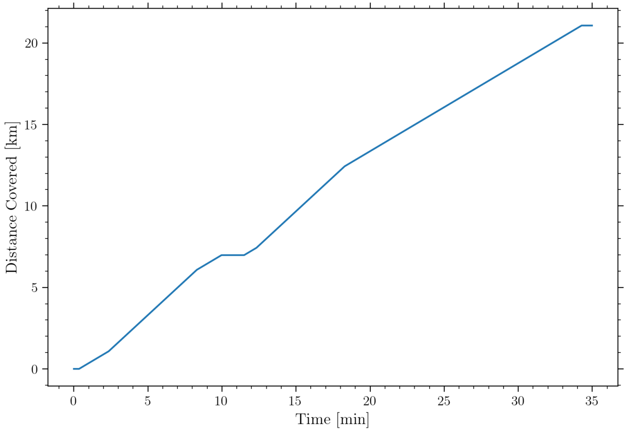


Figure 20: Distance Covered vs Time

Mission C: Fuel limited unsuccessful payload pickup mission

We assume all the same rates as in Mission B. The distance covered in the level flight segment is increased to 520 km. This results to fuel running out in the steady flight segment after picking up the payload.

Mission planner test: Mission C plots

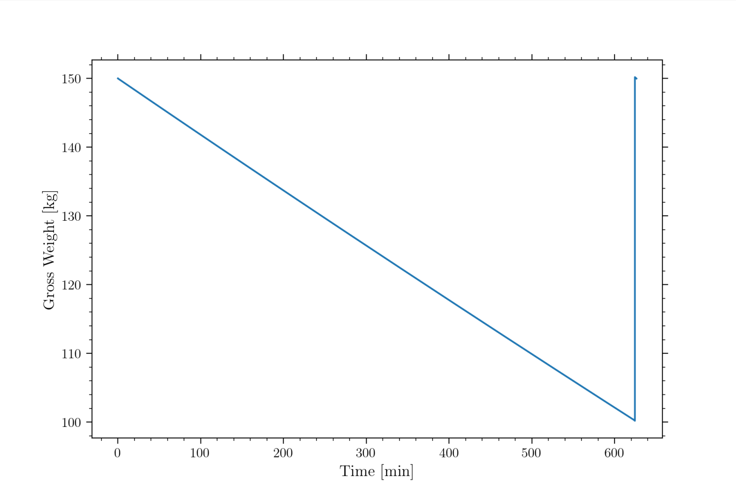


Figure 21: Gross Weight vs Time

Mission planner test: Mission C plots

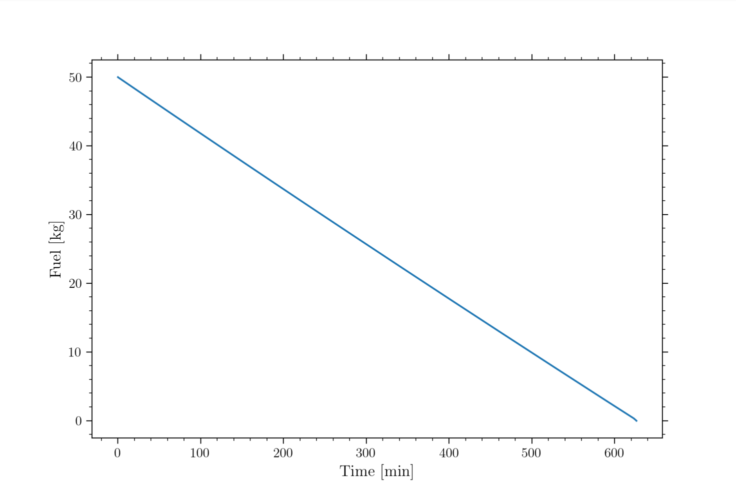


Figure 22: Fuel vs Time

Mission planner test: Mission C plots

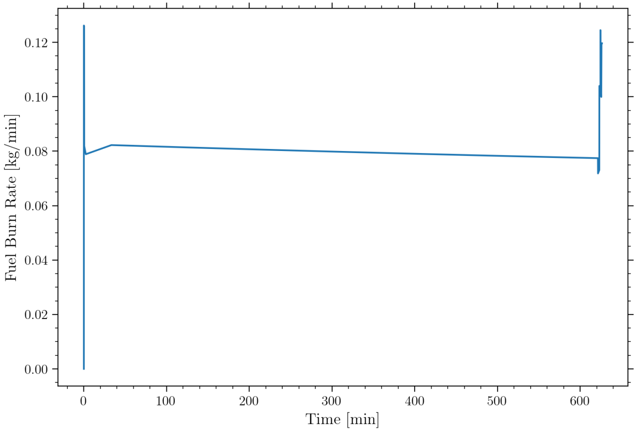


Figure 23: Fuel Burn Rate vs Time

Mission planner test: Mission C plots

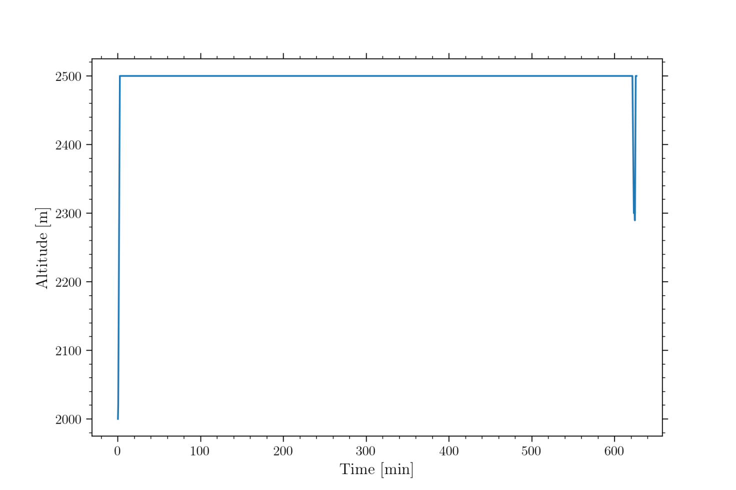


Figure 24: Altitude vs Time

Mission planner test: Mission C plots

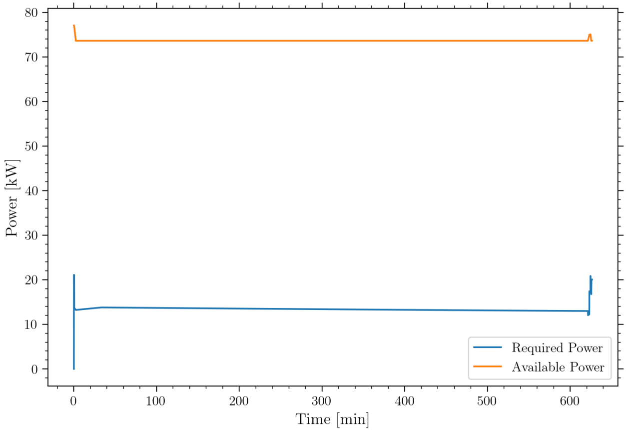


Figure 25: Power vs Time

Mission planner test: Mission C plots

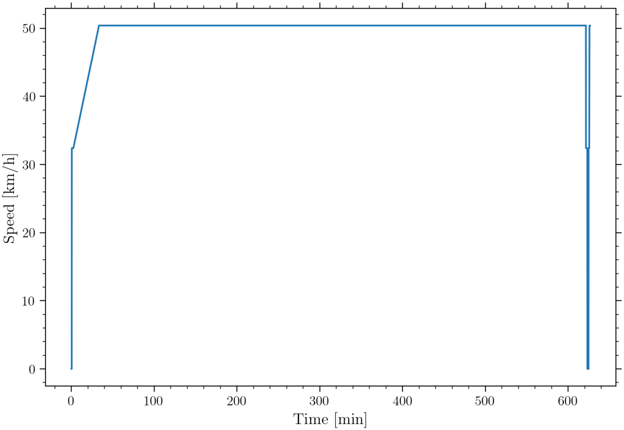


Figure 26: Speed vs Time

Mission planner test: Mission C plots

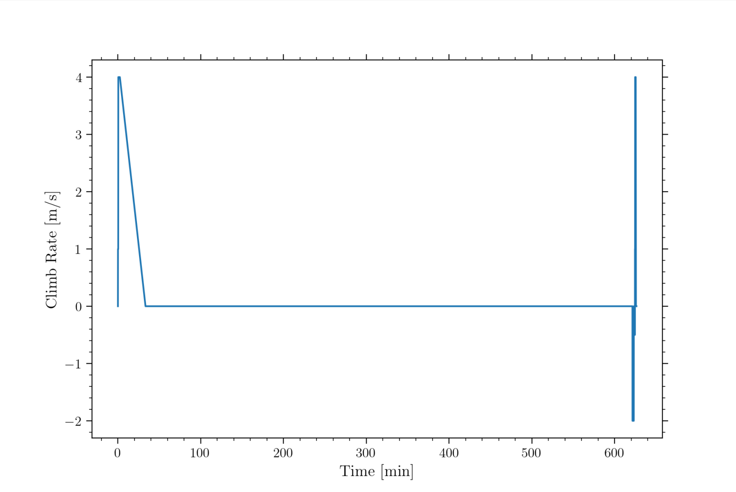


Figure 27: Climb Rate vs Time

Mission planner test: Mission C plots

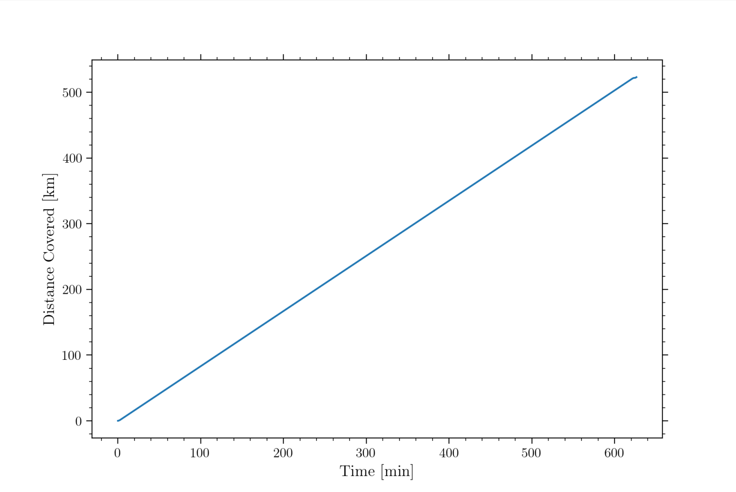


Figure 28: Distance Covered vs Time

Mission planner test

Mission D: Power limited unsuccessful payload drop mission

We assume all the same rates and distances as in Mission A. The helicopter was made to takeoff and perform the mission segments at higher altitudes to check if the power limit is reached. The power limit was not reached even after fliying at 10 km, and going higher did not result in any changes as our ISA table (`data/isa.csv`) is limited to 10 km.

A probable cause for this could be that we had trimmed the helicopter in the 2000-2500 m range to obtain our best fit values of κ, C_{d0} etc. So, the power calculations for higher altitudes wouldnt be accurate. This would require trimming repetitively at higher altitudes, which we did not do due how long it would take to trim each time.

Mission planner test: Mission D plots

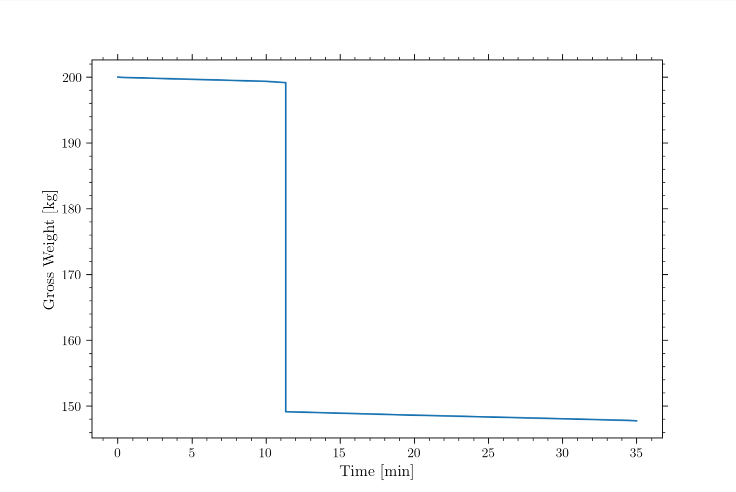


Figure 29: Gross Weight vs Time

Mission planner test: Mission D plots

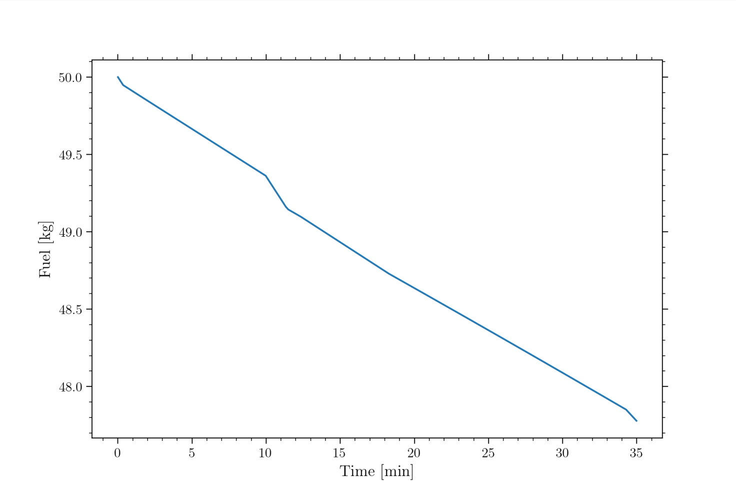


Figure 30: Fuel vs Time

Mission planner test: Mission D plots

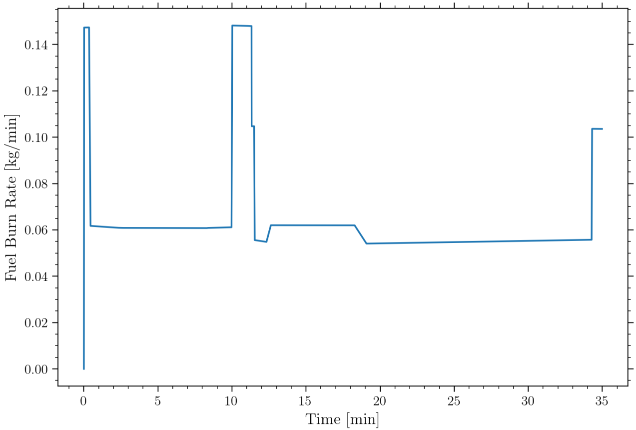


Figure 31: Fuel Burn Rate vs Time

Mission planner test: Mission D plots

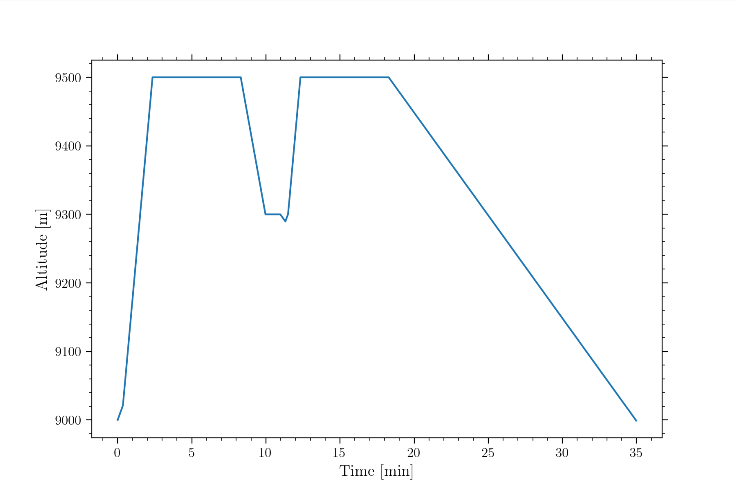


Figure 32: Altitude vs Time

Mission planner test: Mission D plots

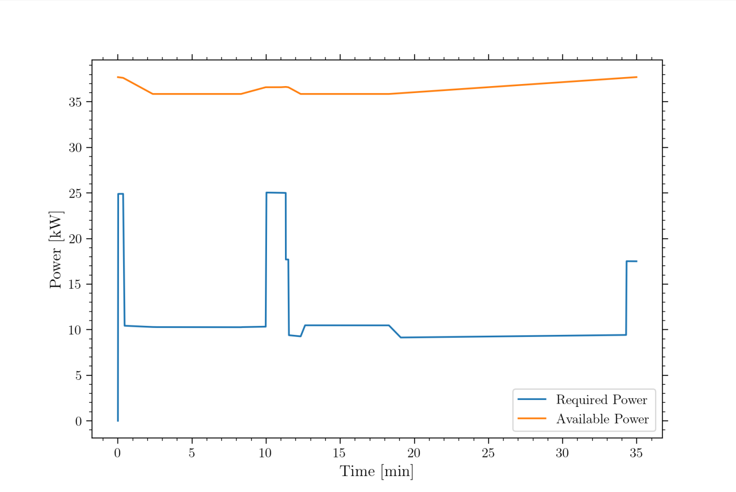


Figure 33: Power vs Time

Mission planner test: Mission D plots

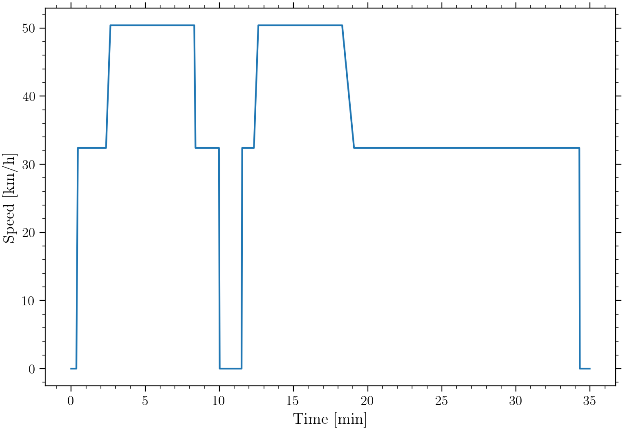


Figure 34: Speed vs Time

Mission planner test: Mission D plots

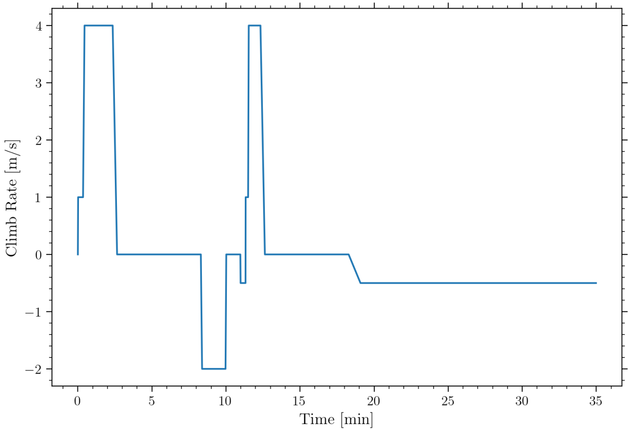


Figure 35: Climb Rate vs Time

Mission planner test: Mission D plots

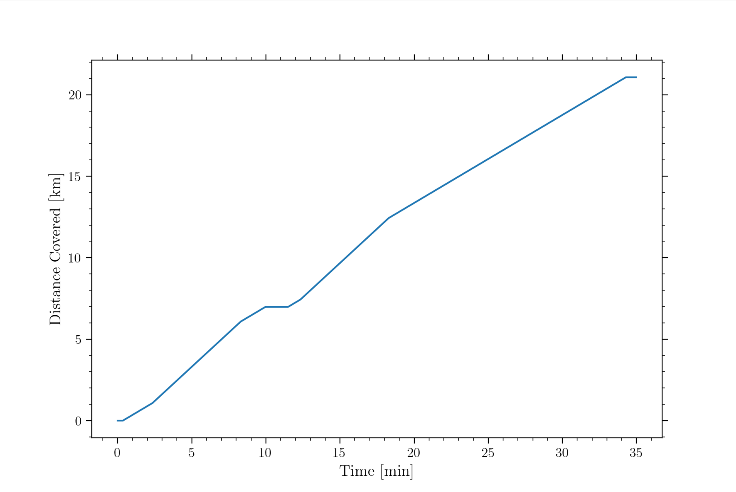


Figure 36: Distance Covered vs Time

Mission planner test - Observations

On analyzing the plots from various missions, we observe how the power varies with altitude, speed, and payload. We also see how the fuel consumption rate varies with time and how the gross weight changes with time. The distance covered by the helicopter is also plotted against time.

Mission A Observations

1. Sudden drop in gross weight observed due to payload drop
2. Fuel burn rate decreases on payload drop (around 13 min)
3. Fuel burn rate also decreases with time in each segment as net weight decreases.
4. Power required increases with altitude, and decrease on payload drop.
5. Distance covered (and fuel weight) is (are) linear with time and increasing (decreasing) as expected.

Mission B Observations

1. Sudden spike in gross weight observed due to payload pickup.
2. Fuel burn rate increases on payload pickup (around 13 min)
3. Power required increases in hover segment, and also on payload pickup.
4. One can also observe that average power required, and average fuel burn rate, before payload pickup is higher than after pickup.

Mission C Observations

1. Fuel runs out in the steady flight segment after picking up the payload (around 650 min).
2. The plots may seem skewed due to the long duration of the level flight segment, but the trends are similar to Mission B.
3. Abrupt changes are numerical issues due to timescale not being fine enough.

Mission D Observations

1. As discussed earlier, the power limit was not reached even at 10 km altitude.
2. We observe that the power required peaks to 25 kW at the payload drop point (around 13 min), but the power available is still higher (at 35 kW).
3. The other trends are similar to Mission A, except for the limits, since the operation altitude is much higher.

References

1. Lecture Notes - AE 667 (Rotary Wing Aerodynamics) by Prof. Dhwanil Shukla