CMM3 Group Design Project

Project Title: Medical Delivery Drone Optimisation

Group Number: Group 22

GitHub Repository URL: https://github.com/tedddant/CMM3-Group-22

Group Member Contributions

Each group member should list 2–3 bullet points summarising their main contributions (technical, organisational, or collaborative).

Name	Student ID	Main Contributions
Edwin Dant	S2474818	- Main contributor to regression-interpolation code. - Minor contributor to root finding code - Set up GitHub repository and drafted the initial report
Ibraheem Hussain	S2476267	-Main contributor to root-finding code and minor contributor to ODE & Regression database research -Involved in research & math at start of project.
Ruoyu Pan	S2494916	- Contribution to ODE numerical model and basic coding. - Helped refining parts of the report scripts
Iustin Tanase	S2549361	-Secondary contributor to the ODE -Helped brainstorm formulas and order of operations at the start of the project

Summary of Group Work

We used Teams, WhatsApp, GitHub and OneDrive to aid in our collaboration. WhatsApp was our primary means of communication due to having faster responses. While important documents and information were shared on teams. Non-code documents were shared using OneDrive, including meeting minutes, initial planning documents, and shortlists. And finally, GitHub was used for code sharing, version control, and reviewing.

We met mostly twice a week- once on Tuesday and once in the seminar on Wednesday. Meeting frequency increased towards the end of the project. We tried pair programming however didn't find it an effective use of time with our level of coding skill. We did carry out reviews of each other's code though.

Tasks were divided amongst the group by joint decision at our in-person meetings and worked mostly effectively. We shared learning and insights throughout the project, often about best practices with using Git and with programming itself. All group members started out using spyder, but some of the group later switched to VS Code due to it being less problematic to run and its better Git integration.

We faced quite tough challenges with the complexity of the system of a drone hovering while trying to consider all the parameters, so after several weeks of research and not feeling like we were making much progress. We spoke to the supervisors and decided to make simplifications to our model to consider mainly the battery discharge of the drone, and the electrical power consumption of the motor-propellers.

Include a list of concrete evidence where possible (does not count towards 250-word limit above):

Meeting Minutes:

Date	Attendees	Notes
24/09	TD,IH,RP,IT	Brainstormed ideas for project, set up communication channels
30/09	TD, IH, RP (IT unable to attend due to timetabling issue)	Decided on Drone optimisation problem. Began filling out project scope ahead of Wednesday session. Still to do: more detailed plan of the mathematical model.
1/10	TD,IH,RP,IT	Worked on project scope
2/10	TD,IH,RP,IT	Finalised and submitted scope
07/10	TD, RP, IT, (IH not present)	Worked on getting to grips with GIT and researching existing drone tests.
08/10	TD, RP, IT, IH	Continued research and discussing ideas
14/10	TD, RP, IT, IH	Discussed our key equations and how to get them to the right level of complexity for the task.
15/10	TD, IT, IH,(RP ill)	Did more research, not much tangible progress made.
21/10	IT,IH,RP (TD had appointment)	Found a useful propeller database and some papers on propeller optimisation.
24/10	TD IH	Met for three hours to make more progress on ODE, came up with a plan of action, more inputs of the ODE
29/10	TD, RP, IT, IH	Spoke to supervisor, now focusing ODE on battery discharge.
4/11	RP, IT, IH (TD had Lab)	Realised we hadn’t been using GitHub correctly so rectified that And looked at each others code so far
5/11	TD, RP, IT, IH	Continued to write code for all three numerical methods and discussed integration between them
12/11	TD, RP, IT, IH	Worked on tidying up repository, continued to draft report, Adjusting code
17/11	TD, RP, IT, IH	Continued report writing
18/11	TD, IT, IH	Continued report writing, final edits to code
19/11	TD, RP, IT, IH	Continued report writing, final edits to code, filled in cover page
20/11	TD, RP, IT, IH	Final edits and submission

As the project progressed we got more familiar with GitHub and began to use commits and branches more effectively, Our early version history/control is quite sparse due to our lack of understanding of Git, we started committing files and using branches later in the project once we understood better. Though we acknowledge we still have some way to go to use best GitHub practices.

More evidence of our consistent communication is available in our Teams chat.

Generative AI Use Declaration

Please check the latest University of Edinburgh guidance on generative AI usage - https://information-services.ed.ac.uk/computing/comms-and-collab/elm/guidance-for-working-with-generative-ai.

AI Statement - "Academic integrity is an underlying principle of research and academic practice. All submitted work is expected to be your own. AI tools (e.g., ELM) should not be used to generate content for this assessment. However, you are allowed to use these tools to identify ideas, key themes, and plan your assessment. You may also use it to improve the clarity of your writing. If you use AI software, you must acknowledge its use in your submission."

Please list below the Generative AI models used in your project:

- ChatGPT, Claude.

Please state the purpose of use (below is an example list, include all that apply and add if needed):

- Brainstorming/outlining
- Concept clarification
- Grammar/spell-check
- Code suggestions/debug hints/tests
- Reference re-formatting
- Latex formatting hints

Introduction

In many parts of the world, access to medical supplies is limited not by availability, but by the ability to transport them into remote areas with poor or unreliable road access. In these cases, where payloads are generally not bulky or heavy i.e. pharmaceuticals, fixed wing drones have already been shown to be highly effective for distribution such as in Rwanda by a company called Zipline [1]. Aerial delivery cuts out reliance on roads entirely and allows fast efficient, package transport to anywhere within a radius of a distribution centre.

We set out to model a small quadcopter drone hovering with a defined payload and battery capacity for a real-world remote drug delivery system. Aiming to optimise the drone for maximum hover time with this representing delivery range in the real world. The greater the range of a quadcopter, the larger area that the distribution centre can serve.

As the project progressed we have adjusted our scope to focus more on battery management, and coefficients of thrust and power - key factors affecting the hover endurance of a drone. This simplification allowed us to model separate aspects of the system while still representing the drone accurately.

System Overview

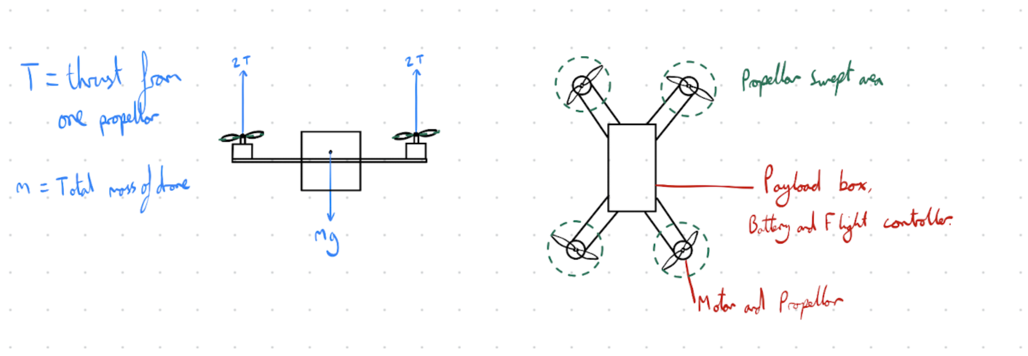

T = thrust from one propellor
m = Total mass of drone
2T
2T
mg
Propellor Swept area
Payload box, Battery and Flight controller
Motor and Propellor

Figure 1: Diagram of drone hovering

Quadcopters are complex systems comprised of 4 counter rotating propellers on each corner of a frame, controlled by a central flight computer with accelerometers, gyroscopes, and GPS. The flight computer feeds instructions to 4 separate electronic speed controllers (ESCs), one for each motor which control the current and voltage supplied to each motor in order to alter the thrust at each corner of the drone. They can be either radio controlled, autonomous or a combination of the two.

In a hover, we want no movement in all 6 degrees of freedom of the drone. This is achieved by having the same rotational speed and thrust output from all four propellers, while having two propellers spinning clockwise, and two spinning anticlockwise, with diagonally opposite propellers spinning the same way.

Factors affecting the Flight time of a drone include; system weight, propeller diameter, propeller efficiency, battery capacity and motor efficiency. We found the physics governing propeller thrust and efficiency, described in this NASA article [2] very difficult to model so we decided not to delve into this too much in this project.

Mathematical Modelling and Numerical Methods

Root Finding for Hover RPM Determination

To determine the hover operating point of the quadcopter, the rotor speed n (rev/s) must satisfy the hovering thrust requirement. For level hover, the total thrust generated by the four propellers is set equal to the total weight of the vehicle:

$$4T(n, D) = W = (m_0 + m_b)g, \tag{1}$$

where D is propeller diameter, m_0 the structural and payload mass, m_b the battery mass, and g gravitational acceleration. Using standard propeller aerodynamics, the thrust per rotor can be expressed as:

$$T(n, D) = \rho\, C_T(D)\, n^2 D^4, \tag{2}$$

where ρ is air density and $C_T(D)$ is the thrust coefficient. Substituting into Eq. (1), the hover condition becomes the nonlinear residual function:

$$f(n) = 4\rho D^4 C_T(D) n^2 - (m_0 + m_b)g. \tag{3}$$

Solving $f(n) = 0$ provides the rotor speed required for static equilibrium. Since this equation is nonlinear in n, analytic solutions are inconvenient, therefore two numerical methods were implemented fora more robust validation: the Bisection Method and the Newton–Raphson Method.

Bisection Method. This is a bracketing method that relies on a sign change in $f(n)$ over an interval $[n_{\text{low}}, n_{\text{high}}]$. The lower bound is taken as $n_{\text{low}} = 0$, and the upper bound is determined using a maximum allowable tip Mach number:

$$n_{\text{high}} = \frac{M_{\text{tip,max}}\, a}{\pi D}, \tag{4}$$

ensuring physically possible rotational speeds. At each iteration, the midpoint is evaluated and the interval is halved based on the sign of $f(n)$. This method guarantees convergence provided the residual is continuous.

Newton–Raphson Method. A derivative-based approach which applies

$$n_{k+1} = n_k - \frac{f(n_k)}{f'(n_k)}, \tag{5}$$

with

$$f'(n) = 8\rho D^4 C_T(D)\, n, \tag{6}$$

yielding locally quadratic convergence when the initial guess is close to the solution. A practically reasonable initial estimate $n_0 = 1000$ rev/s was selected.

Verification and Convergence behavior.

Additionally, a sensitivity analysis was performed across a practical range of propeller diameters (0.22–0.30 m). The percentage difference between the two solutions remained below $2 \times 10^{-6}\%$, confirming that method choice has negligible influence on the final design outcome.

ODE for Energy dissipation and Battery Optimisation

The core of the analysis is the Ordinary Differential Equation (ODE) that models the rate of change of the battery's stored energy (E) over time (t). This rate of change is dictated by the instantaneous electrical power required for the drone to hover (P_{elec}).

$$\frac{dE}{dt} = -P_{\text{elec}} \tag{7}$$

The total electrical power required is the sum of the power needed for lift (P_{lift}) and the constant power consumed by electronics (P_0):

$$P_{\text{elec}} = P_{\text{lift}} + P_0 \tag{8}$$

P_{lift} is calculated on the basis of simplified momentum theory. This theory states that the power scales with the total weight (W) raised to the power of 1.5, corrected by the coefficient k_{losses}, which incorporates system inefficiencies (η_{sys}). The final model uses the total system mass (M_{total}) as a constant for each simulation: $M_{\text{total}} = m_{\text{drone}} + m_{\text{battery}}$.

$$P_{\text{lift}} = k_{\text{losses}}(M_{\text{total}} \cdot g)^{3/2} \tag{9}$$

The coefficient k_{losses} is defined based on the aerodynamic properties of the drone and the combined efficiency of the system:

$$k_{\text{losses}} = \frac{1}{\eta_{\text{sys}} \cdot \sqrt{2\rho A_{\text{total}}}} \cdot g^{-3/2} \tag{10}$$

Substituting this back into the primary ODE yields the final form solved numerically:

$$\frac{dE}{dt} = -\left(k_{\text{losses}}(M_{\text{total}} \cdot g)^{3/2} + P_0\right) \tag{11}$$

Note on Model Evolution: The initial plan included a dynamic mass term ($M_{\text{total}} = m_{\text{fixed}} + \alpha E$), which was abandoned as it implied a non-physical mass reduction in an electric battery. The final model uses the constant M_{total} calculated from the initial battery weight, ensuring physical accuracy.

The final ODE (Equation 11) is a non-stiff Initial Value Problem (IVP) solved using the Python library `scipy.integrate`. The numerical method employed is the Runge-Kutta method of order 5(4) (`RK45`), an accurate and efficient solution path.

- **Initial Condition:** The simulation begins at $t = 0$ with the total available battery energy, $E(0) = E_{\text{battery}}$.
- **Termination Condition:** The ODE integration stops when the remaining energy (E) hits the minimum safe level (E_{min}) dictated by the safety discharge depth.

Interpolation/Regression

We planned to use regression and interpolation to find trends in real world propeller test data, to see if our models produce data with similar trends.

For example, seeing if our root finding identifies a realistic thrust output for a given size and power. The regression/interpolation code begins with a function to access a folder of CSV files and organise into a dictionary of pandas dataframes. The next function interpolates each of the individual propeller dataframes (using UnivariateSpline from scipy.interpolate), to perform cubic spline interpolation to fill in gaps in the data in case we needed to compare exact values produced by our mathematical model. The interpolated data is written to new, separate dataframes in a new dictionary. So that the data is accessible and exportable without altering the original data.

The code also fits regression lines to each of the propeller data sets. Using numpy.polyfit and numpy.poly1d to perform quadratic polynomial regression following the least squares principle. For each propeller, the code fits a quadratic equation to the measured data points for CT and CP versus RPM. It calculates the coefficients and the R^2 value (quality of fit) for each corresponding curve and prints these to the terminal.

Design Analysis

The optimisation sweep determined the following parameters for the configuration that yields the maximum flight time, subject to system power constraints:

Table 1: Optimal Battery Configuration for Maximum Flight Time

Parameter	**Value**	**Unit**
Optimal Capacity (C)	8923	mAh
Battery Mass (m_{battery})	771	g
Total Drone Mass (M_{total})	3.77	kg
Required Power (P_{elec})	788.3	W
Maximum Flight Time	13.2	minutes

The previous model critique stated that the model would predict infinite flight time for larger batteries. However, the final implemented model (Equation 11) incorporates physics-based constraints, which proves the initial critique incorrect. The optimisation success relies on correctly modeling the Energy-vs-Weight trade-off.

1. **The Core Trade-off:** The optimisation demonstrates that increasing capacity provides an Energy benefit (proportional to battery capacity) but incurs a large Weight penalty (where power demand scales as $P \propto M^{1.5}$). The optimal capacity is the point where these two factors balance, maximizing the ratio of storable energy to required power.

2. **Critical Constraint:** The model enforces a hard physical constraint by comparing the calculated P_{elec} for each battery configuration against the drone's maximum surge power ($\text{P}_{\text{MAX\_SURGE}}$). Configurations requiring power greater than this limit are immediately flagged as non-viable (flight time = 0). This step is critical as it models the real-world limits of the motor system, explaining the abrupt collapse of flight time for oversized batteries.

The regression-interpolation script outputs a series of fitted quadratics which could then be used to produce Cp and Ct input values for the root finding equation. The outputs of the code interpolation and regression functions were validated by plotting against the original data and comparing if the values and fitted curves lined up properly.

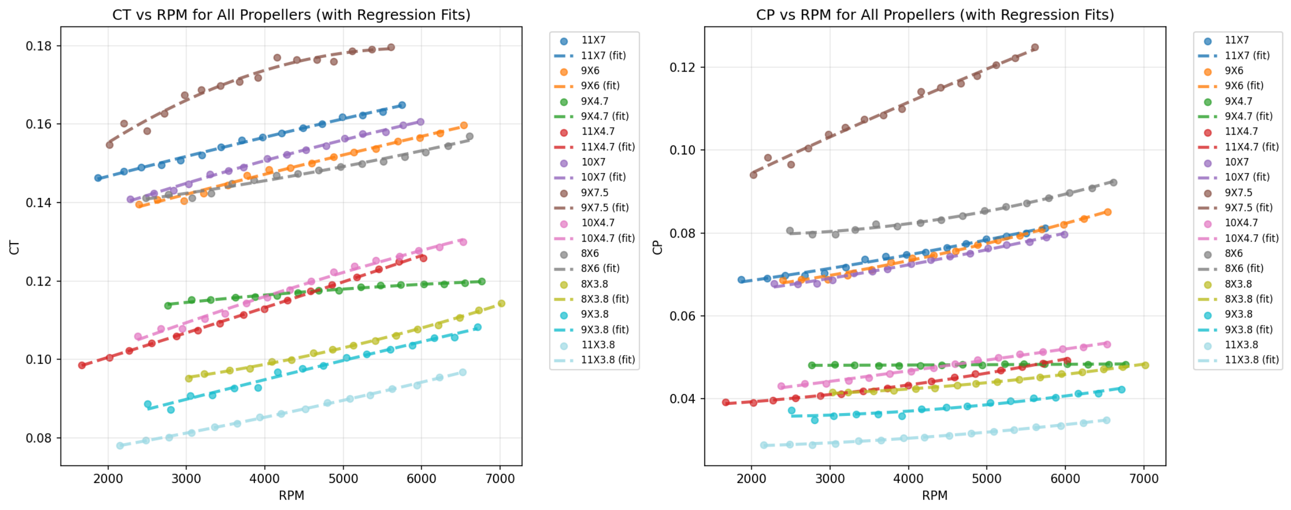


Figure 2: Plots of C_T and C_P against RPM with regression lines fitted. Note: Data labels refer to propeller dimensions in inches (Diameter X Pitch)

Figure 3 shows the predicted hover electrical power requirement as a function of propeller diameter in the range 0.22–0.30 m. As the propeller diameter increases, the model consistently predicts a reduction in hover power. This behaviour is physically expected: enlarging the disc area reduces the induced velocity needed to produce the thrust equal to the vehicle's weight, so less aerodynamic power is required. The reduction is significant across the tested range, with power falling from approximately 215 W at $D = 0.22$ m to around 160 W at $D = 0.30$ m.

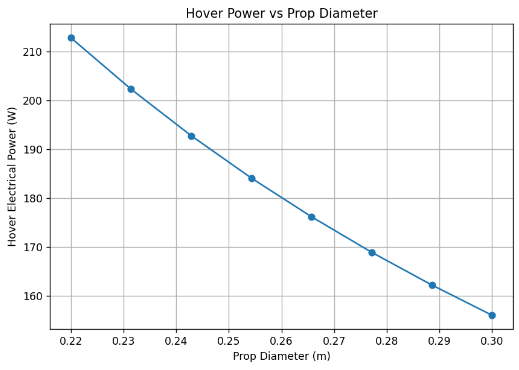
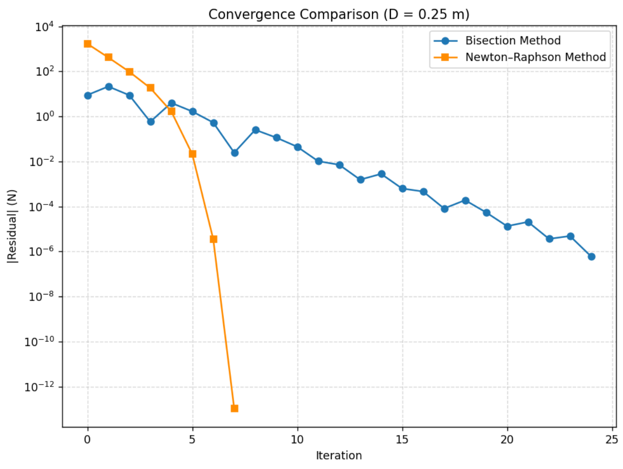


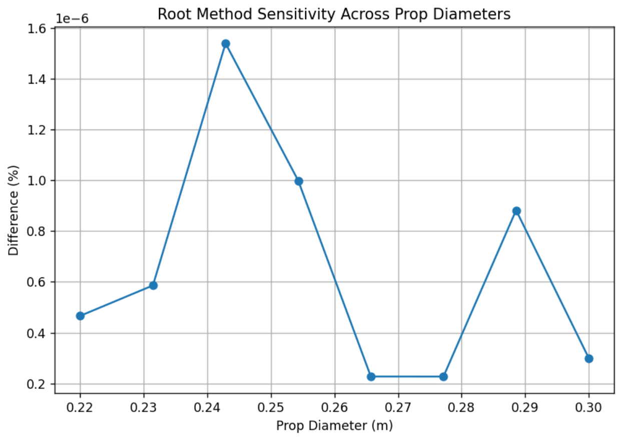
Figure 3: Predicted hover electrical power requirement for propellers with diameters between 0.22 m and 0.30 m. Larger diameters reduce power demand due to increased disc area.

Based on these simulation results, a propeller diameter of 0.25 m was selected for the final design. At this diameter, the predicted hover electrical power is approx. 184 W, corresponding to around 46 W per motor. This level of performance aligns well with commercially available quadcopters of similar total mass (3–4 kg), which gives confidence that the thrust and power levels predicted by the model are realistic. Although larger propellers ($D > 0.26$ m) appear more power-efficient, several practical design considerations justify the selection of $D = 0.25$ m:

A numerical sensitivity test was also performed by solving the hover condition using both the Bisection and Newton–Raphson methods. The two approaches converge to the same hover speed with negligible difference ($< 10^{-6}\%$ across the diameter range). This verifies that the numerical solution is stable.



(a) Convergence for $D = 0.26$ m



(b) Root-Finding Sensitivity Across Diameters

Figure 4: Numerical method comparison for hover RPM solution: (left) residual convergence behaviour; (right) sensitivity of root solutions across propeller diameters.

Conclusions

Overall, our project successfully uses different numerical techniques in an attempt to solve the design problem. Based on our model we selected a propeller diameter of 0.25m and a battery size of 8923 mAh. Our root-finding successfully implemented and validated two different methods (Bisection and Newton-Raphson).

Our ODE model gave us a flight time of 13.2 minutes which doesn't really align with modern drone capabilities. This means that our drone delivery system as it stands would have quite a small delivery radius. However, with some tweaking the flight time could most likely be improved.

While successful in optimisation, the model contains key simplifications that affect absolute precision, therefore there are improvements we could have made given more time.

Static Efficiency Assumption: The system efficiency (η_{sys}) is treated as a fixed constant (0.75×0.70). In reality, efficiency is dynamic, decreasing significantly at high current draws (C-rate effects) and high mechanical loading, leading to an underestimation of required power for very heavy battery configurations.

No Propeller/Motor Specificity: The model relies on averaged efficiency constants rather than inputting real-world propeller coefficient data (C_T/C_P) or motor efficiency curves. Integrating the root-finding analysis directly into the sweep would introduce propeller-specific non-linear effects, improving realism.

Battery Discharge Dynamics: The model neglects voltage sag and internal resistance (I^2R) losses that occur as the battery discharges, slightly underestimating the average power draw over the course of the flight. This would have allowed us to get a more accurate representation of a drone in the real world, hence yielding more realistic results.

Regression data relevance: It would have also been beneficial to perform regression on a dataset which was more relevant to the models which we ended up creating, however since these were developed in parallel to the models which then pivoted to battery optimisation, we didn't have time to rewrite the regression-interpolation code to work with a more well suited dataset. Validation of the interpolation and regression techniques could also be improved by adding different techniques to the script so that results could be compared.

Reynold's Number Simplification: The root finding problem used constants (C_T/C_P), constant aerodynamic coefficients to simplify the model. Since propeller performance varies with RPM and Reynolds number, incorporating these would have made the problem more complex but with a potentially higher accuracy. Moreover, incorporating interpolated propeller data directly into the hover equation would have further aided the realism.

References

[1] *Zipline Drone Delivery & Logistics*. en. URL: `https://www.zipline.com/` (visited on 10/09/2025).

[2] *Propeller Thrust*. en-US. URL: `https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/propeller-thrust/` (visited on 10/02/2025).