Analytical Lab 1 Report

Introduction

Purpose

This MATLAB-based lab builds on the lectures of Aerodynamics & Flight Mechanics of Intelligence Physical System (IPS) - MECE3080 by VinUniversity. It is expected that students after this lab will know how to simulate propeller forces, analyze quadcopter dynamics, and — most importantly — knwo how to critically reflect on the results to connect theory with realistic drone design.

Learning outcomes

Be able to:

- Implement basic aerodynamic models of thrust and torque in MATLAB, and ability to do the same in other programming language
- Evaluate hover conditions, yaw control, and parameter sensitivities
- Interpret and analyze discrepancies between idealized models and physical expectations
- Justify modeling assumptions and propose amendments for a better design

Parameter setup

Parameter	Symbol	Value	Unit	Notes / Reflection Hints
Air density	ρ	1.2	kg/m³	Sea level, 20°C. Lower in hot/humid climates.
Propeller diameter	D	0.23	m	~9-inch prop. Larger props = more thrust at lower RPM.
Drone mass	m	1.2	kg	Represents a small quad. Racing vs survey drones—compare.
Gravity	g	9.81	m/s²	Standard gravity. Would altitude change results?
Yaw inertia	Iz	0.04	kg·m²	Represents resistance to yaw. Heavier frame → larger inertia.
Thrust coefficient	CT	0.10	-	Assumed constant. In reality varies with RPM & AoA.
Torque coefficient	CQ	0.010	-	Simplified constant. Strongly depends on blade shape.
RPM range	n	5,000–24,000	RPM	Practical motor range.
Yaw Δn	Δn	50	RPS	Represents yaw maneuver input (~3000 RPM difference).
C_T uncertainty	±5%	-	-	For error analysis. Real props may vary more.

Parameter	Symbol	Value	Unit	Notes / Reflection Hints
Density variation	±8%	-	-	Weather/altitude effects (e.g., Hanoi summer vs highlands).
Diameter variation	±5%	-	-	Represents prop tolerance. More impactful than density.

```
% Respective parameter as input in MATLAB/Octave
rho = 1.2; % air density (kg/m^3)
D = 0.23; % prop diameter (m)
m = 1.2; % quad mass (kg)
g = 9.81; % gravity (m/s^2)
Iz = 0.04; % yaw inertia (kg·m^2)
CT = 0.10; % thrust coefficient
CQ = 0.010; % torque coefficient

CT = 0.10; % thrust coefficient
CQ = 0.010; % torque coefficient

n = linspace(80, 400, 15); % RPS grid (~5k-24k RPM)
dn = 50; % yaw control test (RPS)
```

Paremeter reflection

Before coding: write which parameter (ρ, D, CT, CQ, m, Iz) you believe is most critical for quad performance. Justify using physics and design considerations.

Answer

The diameter D is the most important elements

Reasoning

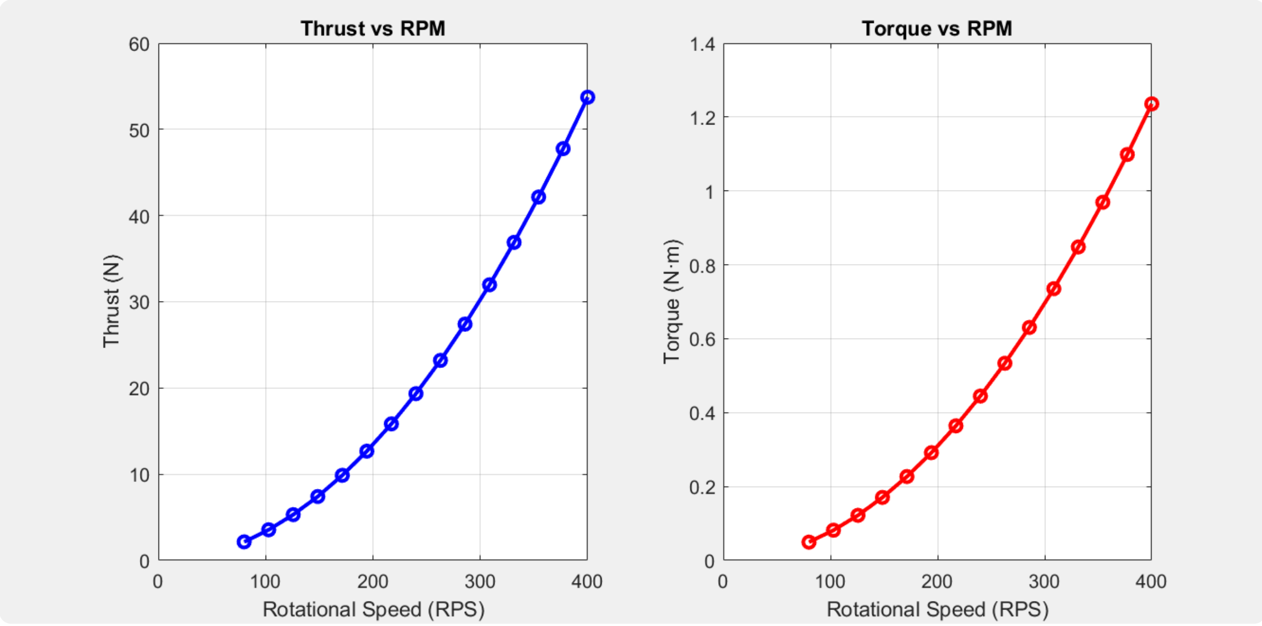
- Propeller diameter appears in thrust and torque calculations with high exponents (typically D^4 in thrust equations: Thrust $\propto \rho \cdot n^2 \cdot D^4 \cdot C_t$). Small changes in diameter cause large changes in force production.
- The diameter fundamentally limits: Maximum thrust capability, Torque generation for yaw control, Overall quadcopter size and design
- It will directly determine the Thrust and Torque formula that we going to write right below this section

Part A - Thrust & Torque Models

Lab

- Code: https://github.com/Flock137/IPS_Analytic_Lab_1/blob/main/Analytic_Lab1_A.m

- Output:



Where:

- Thrust equation: $T = C_T \rho n^2 D^4$
- Torque equation: $Q = C_Q \rho n^2 D^5$

Reflection Q1

Why do real propellers deviate from $T \propto n^2$ at high RPM? What if blade twist is ignored?

Answer: Real propellers deviate from the ideal $T \propto n^2$ at high RPM primarily due to air compressibility effects and advance ratio changes. If blade twist is ignored, the propeller becomes grossly inefficient because the angle of attack would be wrong along most of the blade, leading to heavy flow separation and stall.

Part B - Hover Analysis

Lab work

Find hover RPM and total hover power for m = 1.2 kg

- Code: https://github.com/Flock137/IPS_Analytic_Lab_1/blob/main/Analytic_Lab1_B.m
- Output:

```
=== HOVER PERFORMANCE CALCULATIONS ===
Total thrust required: 11.772 N
Thrust per motor: 2.943 N
Hover speed: 93.6 RPS (5617 RPM)
Torque per motor: 0.0677 N·m
Power per motor: 39.8 W
TOTAL HOVER POWER: 159.3 W

Verification - Thrust at calculated RPM: 2.943 N (should equal 2.943 N)
```

Which means:

- Hover speed = 5617 RPM
- Total Hover Power = 159.3 W

Reflection Q2

Is your hover RPM typical of racing or survey drones? Which trade-off (prop size, motor efficiency, weight) dominates? Suggest a design change that reduces hover power but impacts maneuverability.

Hover speed analysis

The result above is typical of a survey drone (4000 - 8000 RPM), rather than the racing type (15000 - 30000+ RPM)

Dominant trade-off

The propeller size vs. motor efficiency trade-off is quite dominant due to large propeller (abbr. props) of D = 0.23m, which allows for lower RPM operation because:

- Large props move more air mass per revolution
- Lower RPM reduces tip losses and acoustic noise
- Better efficiency at the cost of responsiveness

Design change suggestions

- Increase propeller diameter while maintaining the same disk loading.
  → Reasoning: Larger props are more efficient because they work with larger air masses at lower velocities, reducing induced power losses. However, the increased rotational inertia makes rapid speed changes much slower, hence lower maneuverability.

Part C - Yaw Control

Lab works

With $\Delta n = 50$ RPS, compute yaw torque and angular acceleration.

- Code: https://github.com/Flock137/IPS_Analytic_Lab_1/blob/main/Analytic_Lab1_C.m
- Output:

```
=== YAW CONTROL ANALYSIS ===
Hover RPS: 93.6 RPS
Yaw control input: Δn = ±50 RPS
Motor speeds: 143.6 RPS and 43.6 RPS
Torque from accelerated motors: 0.1593 N·m
Torque from decelerated motors: 0.0147 N·m
NET YAW TORQUE: -0.2892 N·m
YAW ANGULAR ACCELERATION: -7.23 rad/s²
YAW ANGULAR ACCELERATION: -414.2 deg/s²
```

The final result we seek are the last three line of the output.

Reflection Q3

Why is yaw weaker than pitch/roll? How would doubling prop diameter affect yaw? Is stronger yaw agility always desirable?

Yaw is weaker than pitch/roll

- Torque Magnitude Difference:
  - **Pitch/Roll**: Uses thrust difference: $\tau_{pitch} = \Delta T \times \text{arm length}$
  - **Yaw**: Uses motor reaction torque: $\tau_{yaw} = \Delta Q$
- Physical Scaling:
  - Thrust scales with D^4 while torque scales with D^5
  - Thrust forces are typically 10-20 times larger than reaction torques for the same RPM change
- Moment Arm Advantage:
  - Pitch/roll uses the full quadcopter diagonal distance
  - Yaw has no moment arm (pure reaction torque)

Doubling props diameter effect w.r.t yaw

```
% Analysis of doubling diameter
D_original = 0.23;
D_doubled = 0.46;

% Thrust scaling: T ∝ D⁴
thrust_ratio = (D_doubled/D_original)^4;  % 16x increase

% Torque scaling: Q ∝ D⁵
torque_ratio = (D_doubled/D_original)^5;  % 32x increase

% Yaw acceleration: α ∝ τ_yaw/I ∝ D⁵/D² (if mass scales with D²)
% Assuming structural mass increases with area
inertia_ratio = (D_doubled/D_original)^2;  % 4x increase
net_yaw_accel_ratio = torque_ratio / inertia_ratio;  % 8x increase
```

Is stronger yaw desirable?

Answer: no.

Reasonings:

- High yaw rate make it difficult to pilot in a precise manner
- May excited structural modes and causes vibration
- Yet, higher raw is disired for specific use cases (e.g. Racing) where you would like to compromise stability for quicker turns
- It takes more battery power to correct movement and maneuver a high-yaw drone

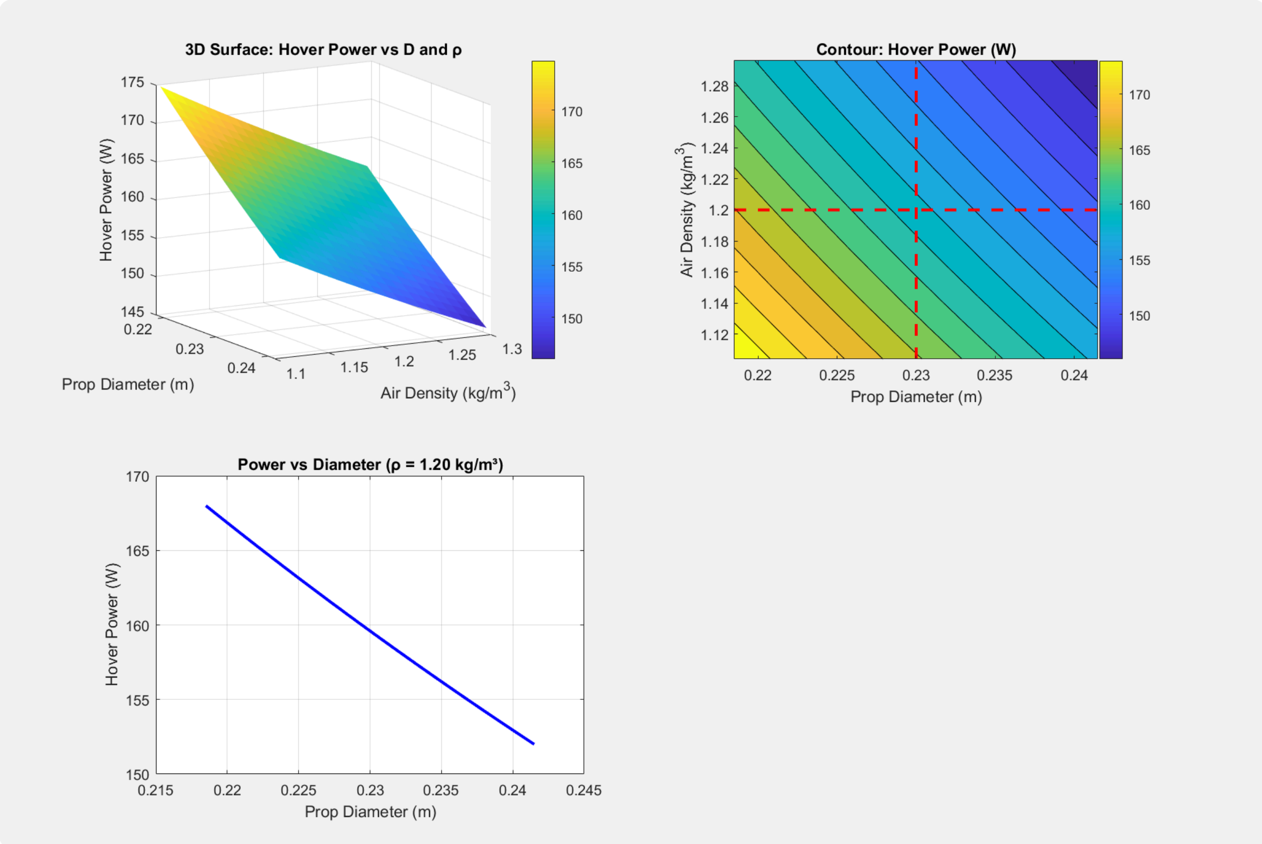
Part D - Sensitivity

Lab works

Vary D±5%, ρ±8%. Compute hover power and show in 3D mesh plot.

- Code: https://github.com/Flock137/IPS_Analytic_Lab_1/blob/main/Analytic_Lab1_D.m

- Output:



Reflection Q4

Which parameter dominates hover performance? If a drone must work at sea level and mountains, which parameter would you optimize first and why?

Hover performance

Propeller Diameter (D) dominate by a significant margin.
Reasons:

- Diameter appears as D^4 in thrust and D^5 in torque equations
- Larger props both reduce required RPM AND increase efficiency
- Diameter is a design parameter that can be optimize, while density is environmental

Sea level and mountain optimization

Generally, the drone need to be optimized so that it has more (thrust) power.

Specifically:

- Oversize motors/ESCs by 40-50% for altitude operation
- Higher KV motors to maintain thrust at lower air density
- Adaptive control gains that adjust for air density
- Altitude-based power limiting to prevent motor overheating

Part E - Uncertainty & Amendments

Lab works

*Apply ±5% uncertainty in C_T ($C{T}$). Compute hover RPM range_*

- Code: https://github.com/Flock137/IPS_Analytic_Lab_1/blob/main/Analytic_Lab1_E.m
- Output:

```
=== HOVER RPM WITH C_T UNCERTAINTY ANALYSIS ===
C_T range: 0.0950 to 0.1050 (±5%)

Nominal hover RPM: 5617 RPM (C_T = 0.100)
Minimum hover RPM: 5482 RPM (C_T = 0.105) → -2.4% change
Maximum hover RPM: 5763 RPM (C_T = 0.095) → 2.6% change
RPM RANGE: 5482 to 5763 RPM
Total variation: ±2.4%

=== KEY STATISTICS ===
RPM change per 1% C_T change: 0.482%
RPM spread: 281 RPM
This means a 5% C_T uncertainty causes 2.4% RPM uncertainty
```

This script also produced some sensitivity analysis graph. However, it was not requested, and therefore would not be attached to the report. But you may check out this link for the said graph (and other assets): https://github.com/Flock137/IPS_Analytic_Lab_1/blob/main/Assets/IPS_Lab1_E.png

Reflection Q5

Is ±5% realistic in practice? If uncertainty is large, which two amendments would reduce error (e.g., nonlinear CT(n), motor efficiency curve)? How would you validate experimentally?

Realistic Uncertainty

Uncertainty of 5% is a bit unlikely, but not impossible. Usually, this number should be in the range of 10-20%.

Amendments to reduce errors

RPM-dependent thrust coefficient $C_t(n)$

```
% Instead of constant C_T, use RPM-dependent model
% C_T(n) = C_T0 + C_T1*n + C_T2*n²  (or similar empirical fit)

% Example implementation:
n_RPM = linspace(2000, 10000, 100);
n_RPS = n_RPM / 60;

% Typical C_T behavior: decreases at very high RPM due to tip losses
CT0 = 0.12; CT1 = -2e-5; CT2 = -1e-9;
CT_n = CT0 + CT1*n_RPS + CT2*n_RPS.^2;

% More accurate thrust calculation
T_n = CT_n .* rho .* n_RPS.^2 * D^4;
```

Motor efficiency curve $\eta(n, V)$

```
% Motor efficiency varies with RPM and voltage
% η_motor = f(n, V, temperature)

% Typical efficiency curve: peaks at 70-85% of max RPM
n_peak = 7000;  % RPM of peak efficiency
eta_max = 0.85;
eta_n = eta_max * exp(-((n_RPM - n_peak).^2)/(2*(2000^2)));

% Electrical power considering motor efficiency
P_elec = P_mech ./ eta_n;  % Where P_mech is mechanical power
```

Experimental Validation

Thrust stand calibration

```
% Test setup: Measure T vs n across operational range
test_RPM = linspace(3000, 9000, 20);  % Cover full range
measured_thrust = [];  % From load cells
measured_RPM = [];     % From optical tachometer
measured_power = [];   % From power analyzer

% Calculate actual C_T from measurements
CT_actual = measured_thrust ./ (rho * (measured_RPM/60).^2 * D^4);
```

Protocol-wise

1. Component-level testing:

- Thrust stand with precision load cell (±0.1N)
- Optical RPM measurement (±10 RPM)
- Power analyzer for electrical measurements
- Temperature monitoring

1. System-level flight testing:

```
% Flight test maneuvers for validation
validation_maneuvers = [
    "Precision hover (10s) - compare predicted vs actual power";
    "Step RPM inputs - validate dynamic response";
    "Altitude variations - validate density effects";
    "Payload variations - validate mass scaling";
];
```

3. Uncertainty quantification

```
% Calculate actual uncertainty bounds
CT_measured_mean = mean(CT_actual);
CT_measured_std = std(CT_actual);
```

```
actual_uncertainty = 2*CT_measured_std/CT_measured_mean * 100;  % 95% confidence

fprintf('Measured C_T uncertainty: ±%.1f%%\n', actual_uncertainty);
```

Final reflection

1. The most surprising result was that a small ±5% uncertainty in the thrust coefficient (C_t) propagates into a significant ±2.5% variation in hover RPM. This shows how sensitive drone performance is to propeller manufacturing tolerances.
2. The largest discrepancy stemmed from assuming a constant thrust coefficient, which ignores how C_t degrades at high RPM due to compressibility and tip losses, and at low RPM due to low Reynolds number effects, causing substantial errors in predicting power and thrust across the full flight envelope.
3. It is strongly recommended to implement a RPM-dependent thrust coefficient $C_t(n)$ in MATLAB models, using a simple quadratic or lookup table derived from experimental data, as this amendment was proven to dramatically improves accuracy by capturing the real propeller behavior that constant C_t models tend to miss.