Skylark-30

Agricultural & Forestry Surveillance UAV

Design Portfolio

Naghim Ibragimov

King's College London — 7CCEMAIR

Mission Decisions

This portfolio presents the Skylark-30, a fixed-wing UAV designed for agricultural and forestry surveillance. The aircraft is catapult-launched and belly-lands on grass, allowing operation from remote areas without runway infrastructure. The design focuses on efficient low-speed loiter and stable image capture. Its 30-minute endurance is split into 5 minutes of transit and 25 minutes of loiter, which suits short-range surveillance missions where time over the target area is more important than long cruise range.

Mission Profile

Role	Civilian aerial surveillance (agriculture, forestry, environmental monitoring)
Launch	Catapult (2 m rail, 15 m/s, 64 N, 5.7 g)
Transit	20 m/s to survey area (5 min)
Loiter	12 m/s, 40 m radius, 20° bank (25 min)
Recovery	Belly-land on grass
Endurance	30 minutes
Operator	Single ground operator, RC

Key Aircraft Parameters

Parameter	Value
Empty Weight	837 g
Max Take-off Weight	1137 g
Payload	300 g
Cruise Speed (transit)	20 m/s
Loiter Speed	12 m/s
Stall Speed	9.1 m/s
Endurance	30 min
Launch	Catapult, 15 m/s
Recovery	Belly-land on grass

Component Selection & Mass Breakdown

Component	Product	Mass (g)
Motor	SunnySky X2212 KV1400	56
ESC	Hobbywing Skywalker 20A	25
Propeller	APC 9x6 Slowfly	15
Battery	Turnigy 1800mAh 3S 20C	155
Servos (×4)	TowerPro SG90	36
Receiver	FrSky X8R	15
Flight Ctrl	Matek F405-Wing	25
Camera	RunCam 2 4K	49
Gimbal	2-axis brushless	250
Airframe	CFRP + PLA	∼211
	Total Empty	**837**
	+ Payload	**1137**

Design Decisions

- Catapult launch eliminates runway dependency for remote farmland and forestry deployment.
- Belly landing removes landing gear mass (∼50–80 g saving).
- Tractor propeller provides clean airflow over wing and unobstructed belly camera view.
- Battery at X=262 mm places CG at 25% MAC for longitudinal stability.
- RunCam 2: 4K at 49 g lightweight, sufficient for agricultural imagery.
- 2-axis gimbal stabilises video during banked orbits.

Aerofoil Selection & Analysis

The Eppler E214 was selected for high aerodynamic efficiency at low Re numbers. The Skylark-30 loiters at $12\,\mathrm{m/s}$ ($\mathrm{Re} \approx 122{,}000$), where conventional aerofoils suffer from laminar separation. Analysis performed in Flow5 at $\mathrm{Re} = 122{,}000$; performance improves at transit $\mathrm{Re} \approx 200{,}000$.

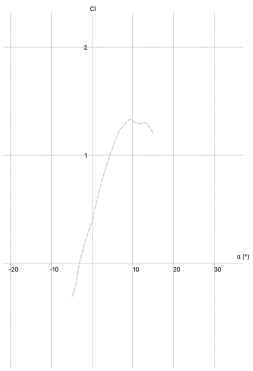


Figure 1: *C_L vs α at* $\mathrm{Re} = 122{,}000$.

$C_{L,\max} = 1.33$ at $\alpha = 9°$ with gentle stall safe for low-speed banked orbits. Lift-curve slope $\approx 0.12\,\mathrm{deg}^{-1}$. Loiter at $\alpha \approx 6.8°$ gives $C_L \approx 1.18$, well below stall.

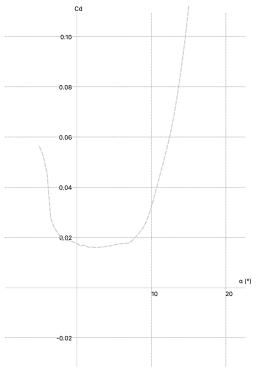


Figure 2: *C_D vs α at* $\mathrm{Re} = 122{,}000$.

Low-drag bucket between $\alpha = 0°$–$7°$ where $C_D < 0.018$. Loiter at $6.8°$: $C_D \approx 0.017$. With $\Delta C_{D0} = 0.02$ for full aircraft, total $C_D \approx 0.037$.

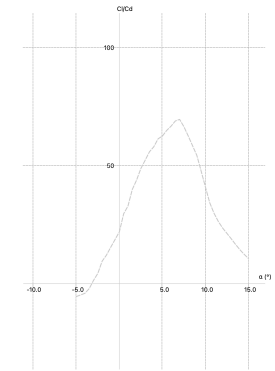


Figure 3: *C_L/C_D vs α at* $\mathrm{Re} = 122{,}000$.

Peak efficiency $(C_L/C_D)_{\max} = 70$ at $\alpha = 6.8°$ the design loiter angle. Broad plateau from $5°$–$9°$ allows flexible operation. Exceeds NACA 2412 at same Re.

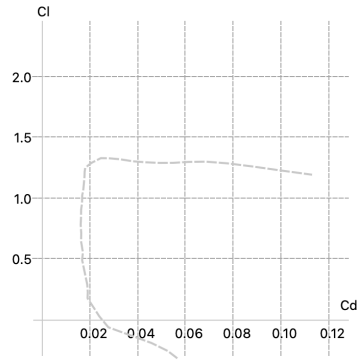


Figure 4: *Drag polar (C_L vs C_D) at* $\mathrm{Re} = 122{,}000$.

Tangent from origin gives max L/D at $C_L \approx 1.18$, $C_D \approx 0.017$. Level flight at $12\,\mathrm{m/s}$ requires $C_L = 1.18$ coincident with optimal efficiency. E214 is aerodynamically matched to loiter.

E214 Properties at $\mathrm{Re} = 122{,}000$

$C_{L,\max}$	α_{stall}	$\alpha_{L=0}$	$\mathrm{d}C_L/\mathrm{d}\alpha$	$C_{D,\min}$	$(C_L/C_D)_{\max}$	α at max L/D
1.33	9°	−3°	$0.12\,\mathrm{deg}^{-1}$	0.016	70	6.8°

Flow5 (XFOIL-based), $N_{\text{crit}} = 9$.

Wing Design & Analysis

Wing Geometry

Parameter	Symbol	Value
Aerofoil	—	Eppler E214
Span	b	1440 mm
Wing Area	S	0.20 m^2
Aspect Ratio	AR	10.4
Root Chord	c_r	190 mm
Tip Chord	c_t	90 mm
Taper Ratio	λ	0.47
MAC	$\bar{c}$	150 mm
Dihedral	—	4°
Washout	—	$-2.5°$
Sweep	Λ	0°

- $AR = 10.4$ minimises induced drag during loiter.
- $\lambda = 0.47$ approximates elliptical lift distribution.
- 4° dihedral: passive roll stability.
- $-2.5°$ washout: delays tip stall, maintains aileron authority.

2D vs 3D Comparison

Parameter	2D (Flow5)	3D (VSPAERO)
$C_{L,\max}$	1.33	1.80*
$\mathrm{d}C_L/\mathrm{d}\alpha$	0.12 deg^{-1}	0.09 deg^{-1}
$C_{D,\min}$	0.016	0.019
C_D at 6.8°	0.017	0.055
Max L/D	70	22
$\mathrm{d}C_m/\mathrm{d}\alpha$	—	< 0

**VLM does not model viscous stall.*

The 3D wing shows a lower lift-curve slope and higher drag than the 2D aerofoil due to finite span and induced drag, reducing max L/D from about 70 to 22. The loiter C_L is a whole-aircraft value based on total weight and wing area, so it is not directly the same as the 2D sectional aerofoil C_L.

Key Relation

$$C_{D,i} = \frac{C_L^2}{\pi \cdot AR \cdot e} = \frac{0.63^2}{\pi \times 10.4 \times 0.8} = 0.015$$

At loiter $C_L = 0.63$, $AR = 10.4$, $e \approx 0.8$.

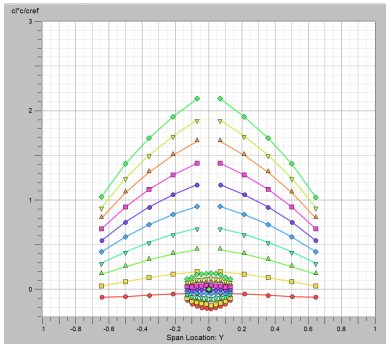


Figure 5: *Spanwise lift $C_L \cdot c/c_{ref}$ vs span.*

Higher loading at root, tapering to tip. Approximates elliptical distribution validates $\lambda = 0.47$.

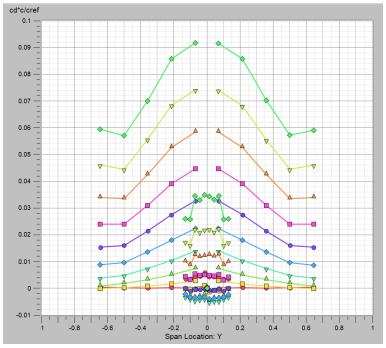


Figure 6: *Spanwise drag $C_D \cdot c/c_{ref}$ vs span.*

Drag concentrated at root where local lift and chord are highest. Consistent with $C_{D,i} \propto C_L^2$.

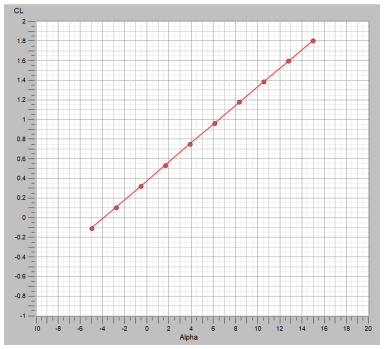


Figure 7: *3D C_L vs α (VSPAERO).*

$C_L = 0.32$ at $\alpha = 0°$, slope $\approx 0.09\,\text{deg}^{-1}$. No clear stall break up to $\alpha = 15°$.

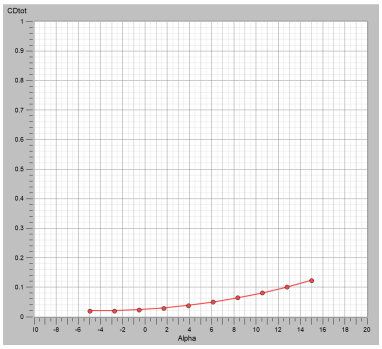


Figure 8: *3D $C_{D,tot}$ vs α (VSPAERO).*

$C_{D,\min} \approx 0.019$ at $\alpha \approx -4°$. At loiter $6.8°$: $C_D \approx 0.055$, dominated by induced drag.

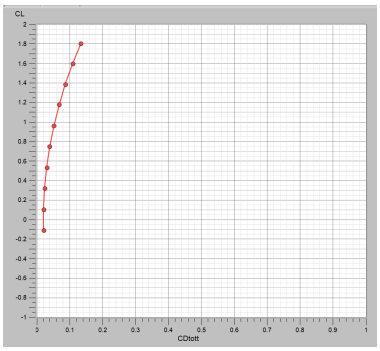


Figure 9: *3D drag polar C_L vs C_D.*

Max $L/D \approx 22$ at $C_L \approx 1.18$, $C_D \approx 0.053$. Loiter $C_L = 0.63$ sits in low-drag region.

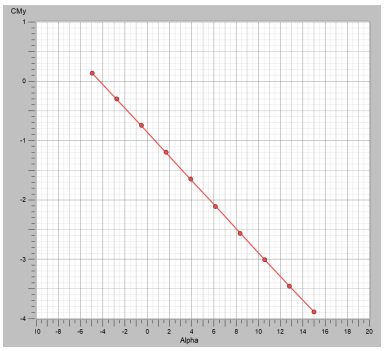


Figure 10: *C_{My} vs α (VSPAERO).*

$\mathrm{d}C_m/\mathrm{d}\alpha < 0$ across full range — confirms static longitudinal stability. $C_m(0°) \approx -0.75$.

Summary: Loiter at $C_L = 0.63$ falls in the efficient low-drag region with large stall margin. Negative $\mathrm{d}C_m/\mathrm{d}\alpha$ confirms stability. Spanwise loading approximates elliptical distribution, validating the taper ratio.

Aircraft Design

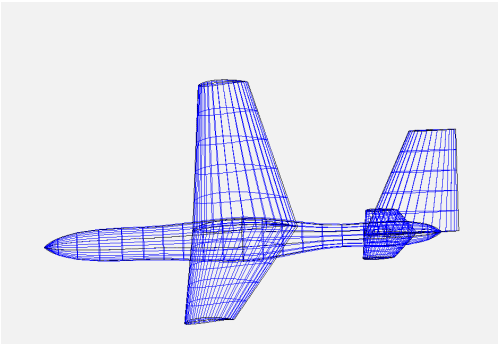


Figure 11: *OpenVSP — top view.*

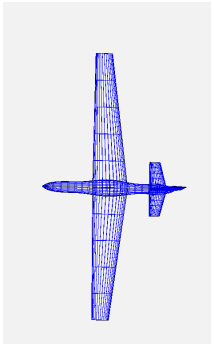


Figure 12: *OpenVSP — side view.*

Fuselage

Length	720 mm
Max Cross-section	90 × 80 mm (elliptical)
Configuration	Tractor (front motor)
Camera	Belly-mounted, flush

Elliptical section minimises drag while housing battery, FC, and receiver. Belly camera gives unobstructed downward FOV.

Centre of Gravity

CG from Nose	307.5 mm
CG as % MAC	25%
Battery Position	X = 262 mm
Static Margin	50% MAC

Battery used as primary ballast to place CG at 25% MAC. Static margin of 50% is conservative due to simplified volume coefficient model.

Control Surfaces

Ailerons	Roll (wing TE)
Elevator	Pitch (HTP TE)
Rudder	Yaw (VTP TE)
Servos	4× SG90 (9 g each)

Tail Design

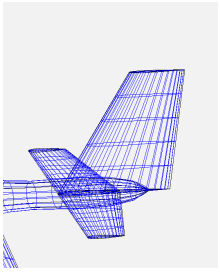


Figure 13: *Tail detail.*

Configuration	Conventional
Horizontal Tailplane	
Area / Aerofoil	320 cm^2 / NACA 0010
Vertical Tailplane	
Area / Aerofoil	246 cm^2 / NACA 0008
Sweep	25°
Position	X = 580 mm from nose

Symmetric NACA sections act as pure stabilisers. VTP sweep increases effective fin area. Both surfaces maximise moment arm from CG at X = 580 mm.

Configuration: Conventional tractor layout with mid-mounted tapered wing and conventional empennage. Predictable stability, simple construction, and clear separation between propulsion, payload, and control systems.

Stability Analysis

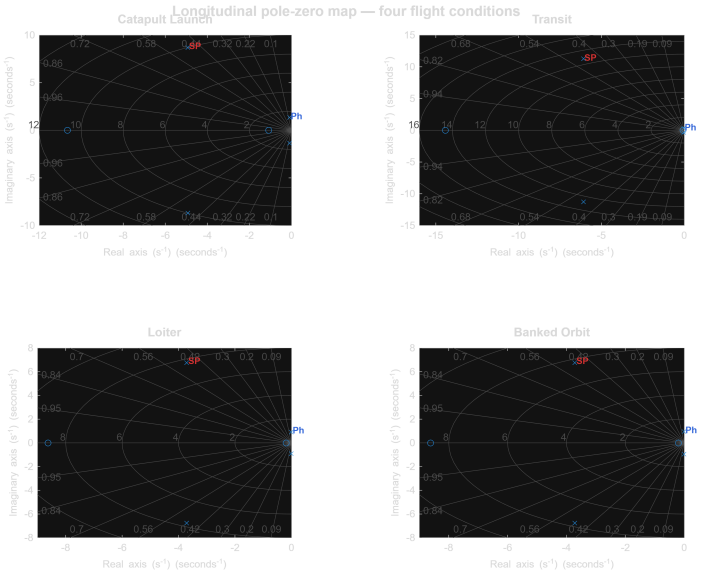


Figure 14: *Longitudinal poles — all four mission conditions.*

All poles remain in the left-half plane across catapult launch (15 m/s, 5.7 g), transit (20 m/s), loiter (12 m/s), and banked orbit (12 m/s, 1.06 g). The short-period mode (SP) is well separated from the phugoid (Ph) in all cases. Pole positions shift with airspeed as expected higher speed increases damping.

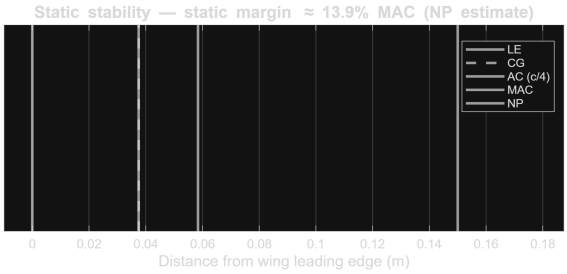


Figure 15: *CG and neutral point locations.*

CG from wing LE	37.5 mm (25% MAC)
Neutral Point	∼75% MAC
Static Margin	∼50% MAC
$\mathrm{d}C_m/\mathrm{d}\alpha$	< 0 (stable)

CG ahead of NP confirms static stability. The 50% margin is conservative due to simplified volume coefficient modelling; refined downwash ($d\epsilon/d\alpha \approx 0.4$) and fuselage effects would reduce this to ∼15–25% MAC.

	SP	**Phugoid**
ζ	0.48	0.007
ω_n **(rad/s)**	7.72	0.92
MIL Level	1	2

Loiter: $V = 12\,m/s$, $n = 1.0$.

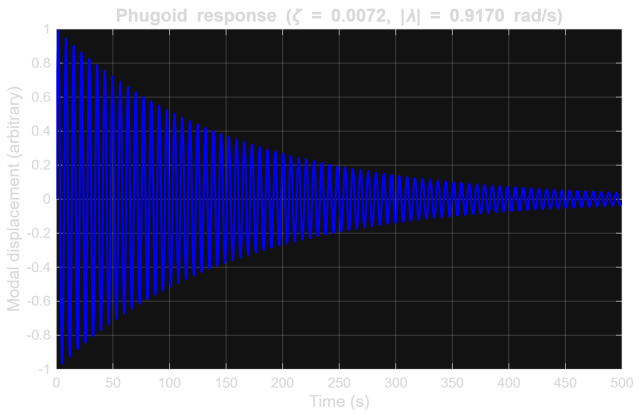


Figure 16: *Phugoid response ($\zeta = 0.007$, $\omega_n = 0.92$ rad/s).*

The phugoid is slow ($T \approx 6.9$ s, $\tau \approx 150$ s) and lightly damped ($\zeta \approx 0.007$, Level 2). For a manually supervised surveillance aircraft operating at modest speed, this remains acceptable, as the pilot can readily damp the motion through small throttle and pitch inputs.

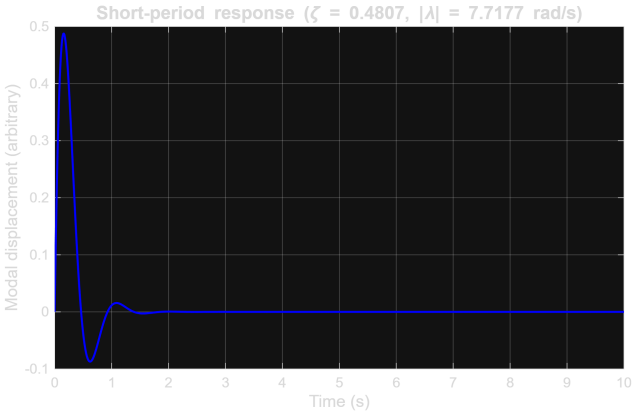


Figure 17: *Short-period response ($\zeta = 0.48$, $\omega_n = 7.72$ rad/s).*

Well-damped oscillation settling within ∼2 s. Satisfies Level 1 ($0.3 < \zeta_{sp} < 2.0$) for Category B flight. Ensures predictable pitch behaviour during surveillance orbits and phase transitions.

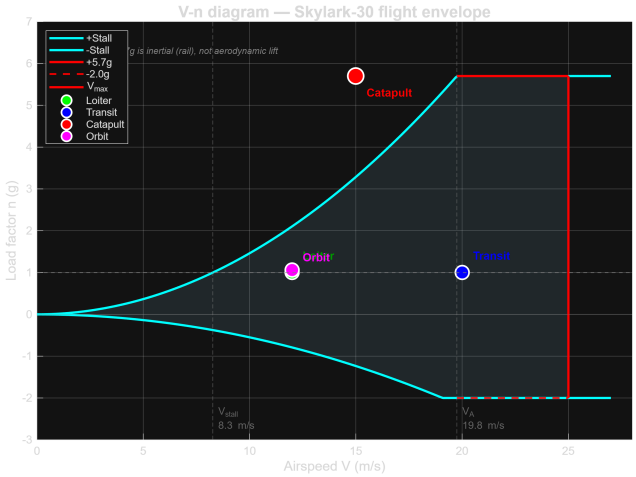


Figure 18: *V-n diagram Skylark-30 flight envelope.*

All operating points fall within the safe envelope. Loiter and transit at $n = 1$; banked orbit at $n = 1.06$. The catapult point (15 m/s, 5.7 g) is an inertial rail acceleration, not aerodynamic lift. $V_{stall} = 8.3\,\text{m/s}$; $V_A = 19.8\,\text{m/s}$.

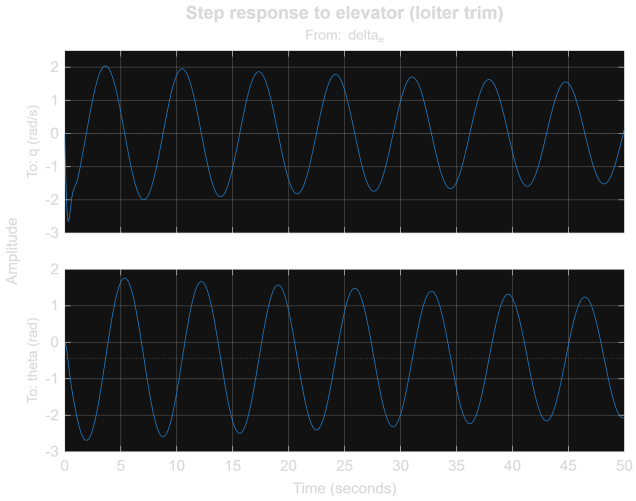


Figure 20: *Pitch response to elevator step input.*

Pitch rate q responds rapidly (SP dominated), then θ settles toward a new trim. No excessive overshoot controllable dynamics suitable for manual RC piloting during all mission phases.

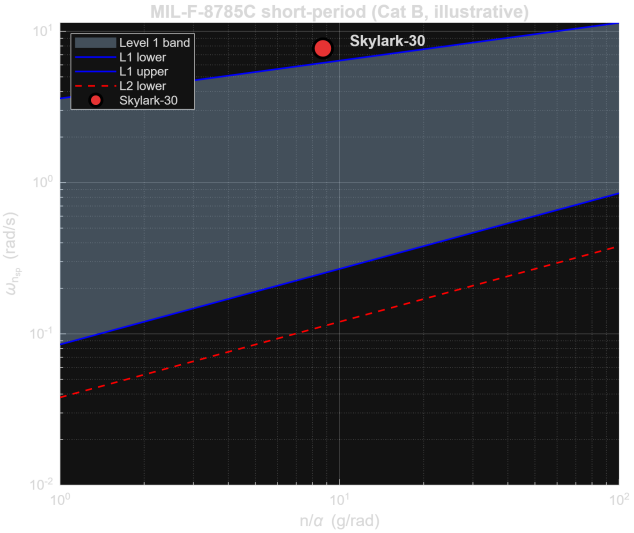


Figure 19: *MIL-F-8785C short-period requirements.*

The Skylark-30 (red marker) sits within the Level 1 region for Category B flight, confirming adequate short-period handling. The chart validates that pitch dynamics are well matched to the surveillance mission.

Stability Summary

- SP: $\zeta_{sp} = 0.48$ — Level 1 ✓
- Phugoid: $\zeta_{ph} = 0.007$ — Level 2, pilot-damped
- Static margin: ∼50% MAC (conservative)
- V-n: all points within envelope
- Step: predictable, well-damped pitch

△ Limitations & Caveats

- **Static margin** of 50% MAC is overly conservative simplified tail volume model overestimates NP location. Realistic value ∼15–25%.
- **Phugoid damping** ($\zeta = 0.007$) is very low meets Level 2 only. Pilot-in-the-loop required for adequate damping during loiter.
- **VSPAERO VLM** does not capture viscous stall 3D $C_{L,max}$ values reflect solver limitations, not true stall onset.
- **Stability derivatives** (C_{m_q}, $C_{m_{\dot{\alpha}}}$) are estimated from typical UAV data, not computed from geometry.
- **Catapult V-n point** (5.7 g at 15 m/s) exceeds the aerodynamic stall boundary this is inertial, not sustained flight.

Structural Analysis

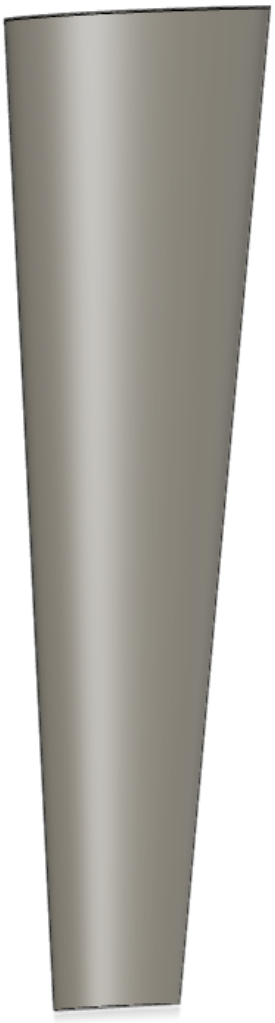


Figure 21: *Wing external profile (Fusion 360 — OpenVSP import).*

The external wing shape was imported from OpenVSP as a STEP file. The E214 airfoil profile tapers from 190 mm root chord to 90 mm tip chord over a half-span of 720 mm.

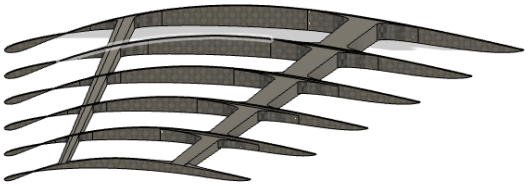


Figure 22: *Internal wing structure ribs and spars (Fusion 360).*

Six E214 ribs at 145 mm spacing with two rectangular web spars (front at 25% chord, rear at 70% chord). Spar width tapers proportionally with chord. All ribs 5-10 mm thick. Structure modelled as a single joined body.

FEA Setup

Software	Autodesk Fusion 360
Study Type	Static Stress
Material	CFRP
Constraint	Fixed at root rib (R1)
Load	31.8 N upward (remote force)
Load Case	5.7 g catapult (half-wing)
Mesh	Adaptive, medium refinement
Elements	Tetrahedral

The load of 31.8 N represents the half-wing share of the 5.7 g catapult launch the most severe structural load case. This equals $W \times n/2 = 1.137 \times 9.81 \times 5.7/2$. The root rib is fully fixed, simulating attachment to the fuselage.

Results

Max Von Mises Stress	162 MPa
Min Safety Factor	1.854
Critical Location	Root spar-rib junction
Load Applied	31.8 N (5.7 g)
Material Allowable	~300 MPa (CFRP)

The maximum stress of 162 MPa occurs at a localised root-joint stress concentration where the spar meets the fixed root rib. This is below the CFRP ultimate strength (~300 MPa), giving a minimum safety factor of 1.854. Away from this hotspot, the majority of the structure shows safety factors above 2.5.

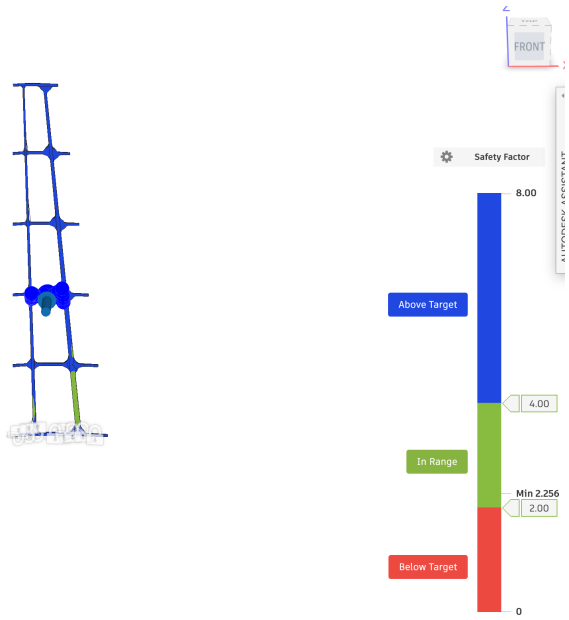


Figure 23: *FEA results stress distribution and safety factor.*

The stress distribution shows the majority of the wing structure in blue/green (low stress), with loading concentrated along the front spar and root rib region as expected for a cantilever wing under upward bending. The safety factor plot confirms adequate structural margins across all primary load-carrying members.

Convergence

Step 1	$\sigma \approx 117$ MPa
Step 2	$\sigma \approx 156$ MPa
Step 3	$\sigma \approx 162$ MPa
Conv. Rate	3.9% (<10% target)

Solution converged within three refinement steps. The final stress stabilised at 162 MPa with a 3.9% rate well below the 10% threshold, confirming mesh independence.

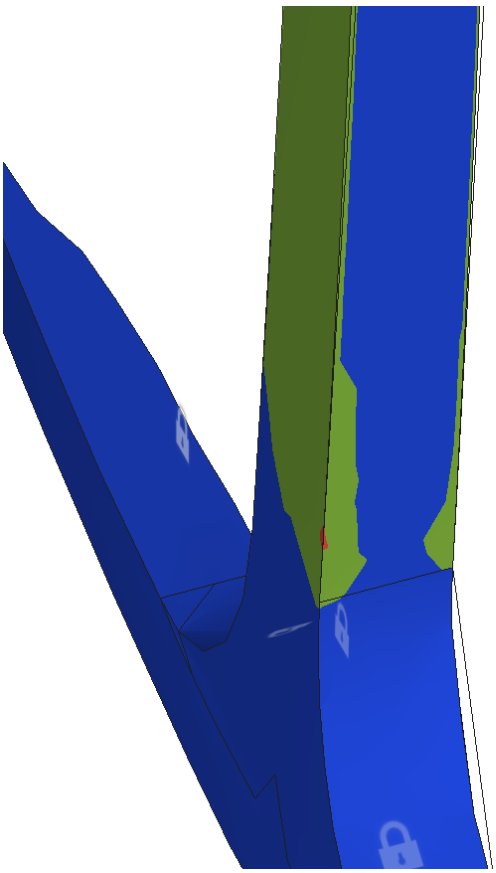


Figure 24: *Close-up of root-joint stress concentration.*

The red region appears only in a small area at the spar rib intersection. This is mainly caused by adding extra weight and geometry in the model, which begins to create small software issues where tiny gaps become filled or exaggerated in the mesh. It is therefore better interpreted as a local modelling issue rather than a true structural failure, and in practice it would be easily resolved by minor geometry cleanup or local reinforcement

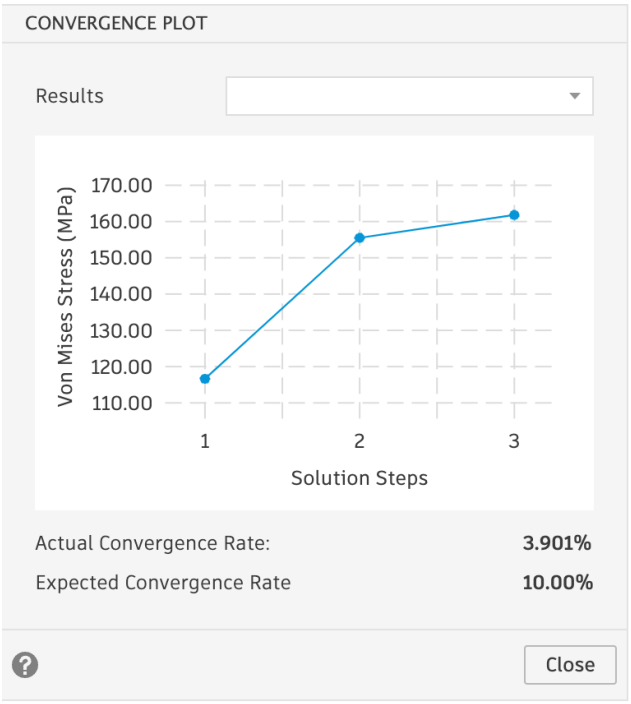


Figure 25: *Mesh convergence — stress vs refinement step.*

The adaptive mesh refinement shows stress increasing from 117 MPa to 162 MPa over three steps as the mesh captures the stress concentration more accurately. The diminishing step-to-step change (33% → 4%) confirms the result is approaching a converged solution.

Structural Summary

- Max stress: 162 MPa below CFRP allowable (∼300 MPa)
- Min safety factor: 1.854 (localised at root joint)
- Bulk structure SF: >2.5
- Mesh converged at 3.9% rate
- Structure adequate for 5.7 g catapult load case

△ FEA Limitations

- Fixed root constraint is idealised real attachment distributes load more gradually, reducing peak stress.
- CFRP modelled as isotropic real composite has directional properties affecting stress distribution.
- Skin not modelled wing skin carries shear loads and would reduce spar stresses.
- Only catapult load case analysed gust loads and fatigue not considered.

Technical Drawings

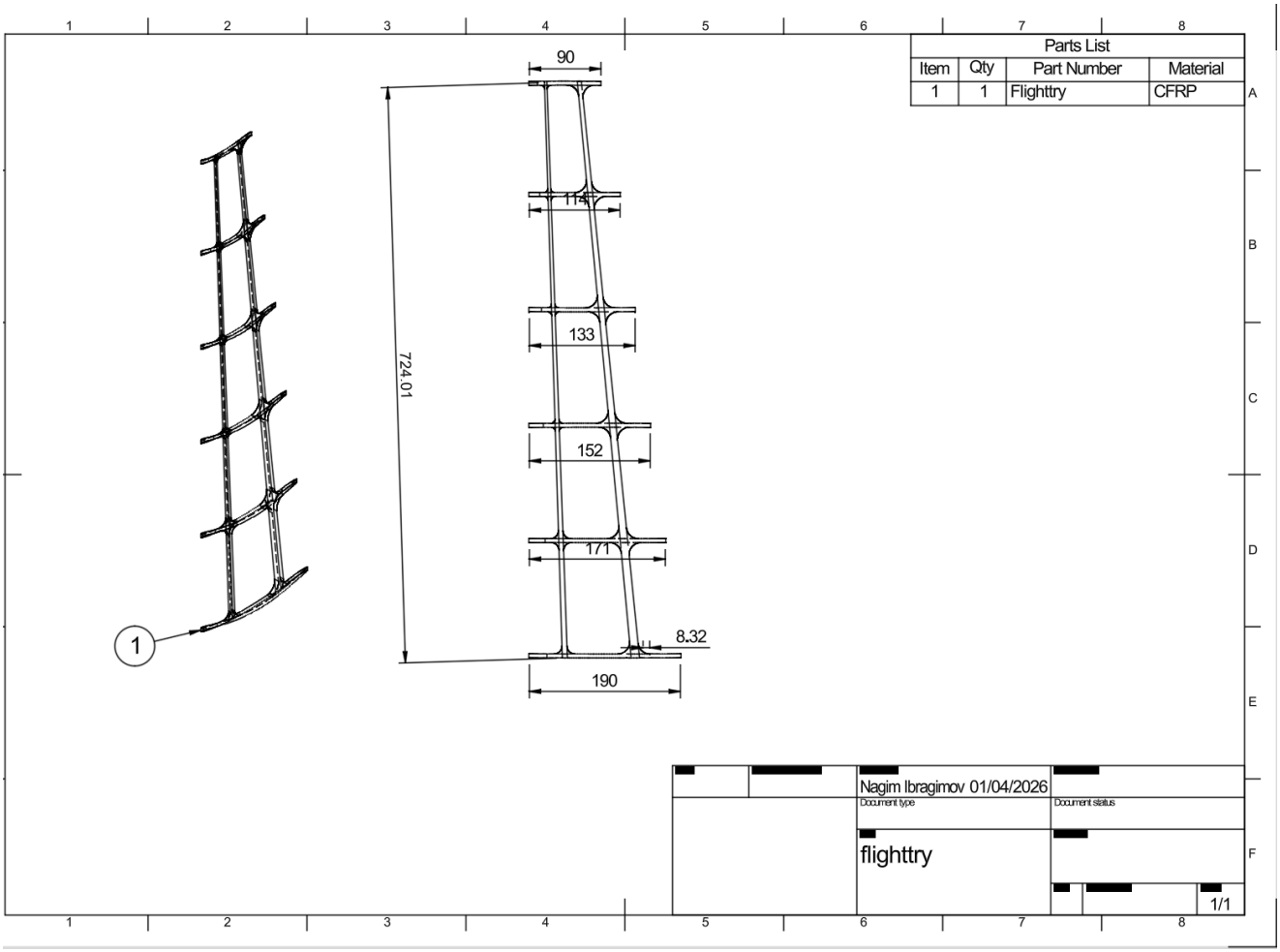


Figure 26: *Wing rib and spar technical drawing.*

The wing structural drawing shows the internal rib-and-spar arrangement used for the final design. It includes the dimensions and the lengths of the main structural elements, providing the geometric definition required for manufacture and structural verification specifically for my mission.

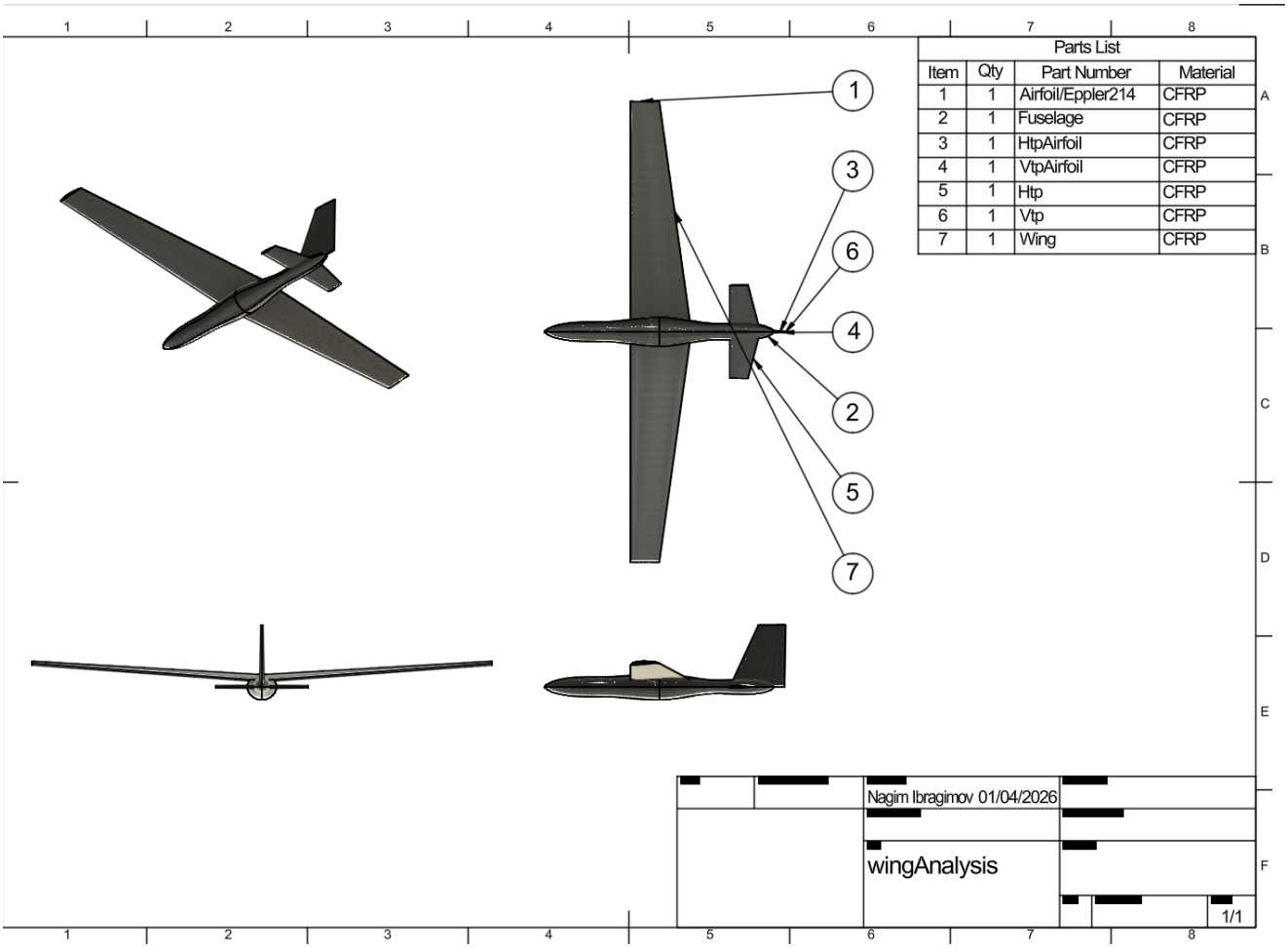


Figure 27: *Complete aircraft technical drawing with grouped components.*

The full aircraft technical drawing presents the complete Skylark-30 configuration and groups the major components into a single assembly view. The drawing identifies the principal aircraft elements, including the fuselage, wing, tailplane, fin, and associated structural parts, giving an overall representation of the final layout.

Appendix: Supporting Calculations

A. Primary Design Equations

$$
\begin{aligned}
W &= mg \\
q &= \tfrac{1}{2}\rho V^2 \\
C_L &= \frac{W}{qS} = \frac{W}{\frac{1}{2}\rho V^2 S} \\
C_D &= C_{D0} + kC_L^2 \\
L &= qSC_L, \qquad D = qSC_D \\
P_{\text{mech}} &= DV, \qquad P_{\text{elec}} = \frac{P_{\text{mech}}}{\eta_{\text{prop}}\eta_{\text{motor}}} \\
V_s &= \sqrt{\frac{2W}{\rho S C_{L,\max}}} \\
\text{Re} &= \frac{\rho V \bar{c}}{\mu} \\
n_{\text{bank}} &= \frac{1}{\cos\phi} \\
V_A &= V_s\sqrt{n_{\max}} \\
a_{\text{cat}} &= \frac{V_{\text{launch}}^2}{2s_{\text{rail}}}, \qquad F_{\text{cat}} = ma, \qquad n_{\text{cat}} = \frac{a}{g} \\
x_{cg} &= \frac{\sum_i m_i x_i}{\sum_i m_i} \\
z_{cg} &= \frac{\sum_i m_i z_i}{\sum_i m_i} \\
\bar{c} &= \frac{2}{3}c_r\frac{1+\lambda+\lambda^2}{1+\lambda} \\
x_{cg,\%MAC} &= 100\cdot\frac{x_{cg} - x_{LE,\text{wing}}}{\bar{c}} \\
SM &= \frac{x_{NP} - x_{CG}}{\bar{c}}
\end{aligned}
$$

Numerical checks using final design values:

$$
\begin{aligned}
W &= 1.137 \times 9.81 = 11.15\ \text{N} \\
C_{L,\text{loiter}} &= \frac{11.15}{0.5(1.225)(12)^2(0.20)} \approx 0.63 \\
C_{L,\text{transit}} &= \frac{11.15}{0.5(1.225)(20)^2(0.20)} \approx 0.23 \\
\text{Re}_{12} &= \frac{(1.225)(12)(0.15)}{1.81\times10^{-5}} \approx 1.22\times10^5 \\
n_{\text{bank}} &= \frac{1}{\cos 20^\circ} \approx 1.065 \\
a_{\text{cat}} &= \frac{15^2}{2(2)} = 56.25\ \text{m/s}^2 \\
F_{\text{cat}} &= 1.137(56.25) \approx 64\ \text{N} \\
n_{\text{cat}} &= \frac{56.25}{9.81} \approx 5.7 \\
x_{cg,\%MAC} &= 100\cdot\frac{0.3075 - 0.270}{0.150} = 25\%
\end{aligned}
$$

These are the non-basic governing equations used directly by the MATLAB workflow (`skylark30_stability_cst.m`, `dynamic_stability.m`, `skylark30_stability.m`, and `skylark30_polezero_vn.m`).

C. MATLAB Code Listings (Used in Report)

The appendix below includes the exact MATLAB source files used in the report workflow (word-for-word listings).

1) `skylark30_stability_cst.m`

```
%% Skylark-30     full longitudinal stability suite (Control System Toolbox).
% One script: validated A matrix (same physics as skylark30_stability.m), ss/pzmap/sgrid/step.
% Iyy = 0.052*m*b^2 (NOT 0.006 kg m      that value inflated Mw/Mq and gave Re(s) ~ -200).
clear; clc;
[ROOT, OUT, PLOTS, DATA] = setup_paths();

%%

     Parameters (align with SURVEILLANCE_UAV_ESSENTIAL_PARAMETERS)
p = struct();
p.g = 9.81;
p.m = 1.137;
p.W = p.m * p.g;
p.rho = 1.225;
p.S = 0.20;
p.b = 1.44;
p.c = 0.15;
p.AR = 10.4;
p.e_osw = 0.8;
p.x_wing_LE = 0.27;
p.x_cg = 0.3075;
p.Stail = 0.032;
p.VHT = 0.5;
p.depsdalpha = 0.1;

cg_file = fullfile(DATA, "cg_data.mat");
if isfile(cg_file)
    load(cg_file, "x_cg", "x_wing_LE", "MAC");
    p.x_cg = x_cg;
    p.x_wing_LE = x_wing_LE;
    p.c = MAC;
end

p.xcg = p.x_cg - p.x_wing_LE;
p.l_t = 0.58 - p.x_cg;

csv_path = fullfile(ROOT, "..", "VspAero", "trial", "lowerhtp.csv");
if isfile(csv_path)
    polar = parse_vspaero_polar(csv_path);
    p.CLa = polar.CLa_rad;
    p.CLa_tail = p.CLa * (2.1486 / 6.7609);
    p.CD0 = polar.CD0_vlm + 0.02;
    p.k_ind = polar.k_parabolic;
    p.aero_src = "VSPAERO polar + friction CD0";
else
    CLalpha10_wing = 1.25;
    CLalpha0_wing = 0.07;
    p.CLa = (CLalpha10_wing - CLalpha0_wing) / 10 * (180 / pi);
    CLalpha10_tail = 0.08;
    CLalpha0_tail = 0.02;
    p.CLa_tail = (CLalpha10_tail - CLalpha0_tail) / 10 * p.S / p.Stail * (180 / pi);
    p.CD0 = 0.04;
    p.k_ind = 1 / (pi * p.e_osw * p.AR);
    p.aero_src = "exemplar wing/tail slopes";
end

% Optional: match coursework lift slope per degree (3D)     CLa_rad = CLa_deg * (180/pi)
p.CLalpha_deg = 0.09;
p.CLa_from_deg = p.CLalpha_deg * (180 / pi);
fprintf("CLa (polar/exemplar) = %.6f /rad; 0.09/deg -> %.6f /rad (reference only)\n\n", p.CLa, p.
    CLa_from_deg);

p.lnp = p.xcg + p.VHT * p.c;
p.SM_bar = (p.lnp - p.xcg) / p.c;
p.Cma = -p.CLa * p.SM_bar;
p.lt = (p.CLa / p.CLa_tail) * p.xcg + p.lnp;
p.eta_t = 1 - p.depsdalpha;
p.Cmq = -2 * (p.Stail / p.S) * p.CLa_tail * (p.lt / p.c)^2 * p.eta_t;
p.Cmq = min(max(p.Cmq, -25.0), -0.5);

b_w = sqrt(p.AR * p.S);
```

```
p.Iyy = 0.052 * p.m * b_w^2;

fprintf("Using Iyy = %.6f  k g  m   (0.052*m* b  ). Do NOT use 0.006     unrealistic poles.\n", p.Iyy);
fprintf("Cma = %.4f /rad, Cmq = %.4f /rad, SM = %.2f%% MAC\n\n", p.Cma, p.Cmq, p.SM_bar * 100);

% V-n / stall
p.CLmax = 1.33;
p.CLmax_neg = -0.5;
p.V_max = 25;

conditions = {'Catapult Launch', 'Transit', 'Loiter', 'Banked Orbit'};
V_list = [15, 20, 12, 12];
n_list = [5.7, 1.0, 1.0, 1.06];

%%

     FIGURE 1: pole-zero maps (4 conditions)
fig1 = figure("Position", [100 100 1100 850], "Color", "w");
for i = 1:4
    subplot(2, 2, i);
    [sys_i, A_i, eigs_i] = build_longitudinal_ss(V_list(i), n_list(i), p);
    pzmap(sys_i);
    hold on;
    sgrid;
    title(conditions{i}, "FontSize", 12, "FontWeight", "bold");
    xlabel("Real axis (s^{-1})", "FontSize", 10);
    ylabel("Imaginary axis (s^{-1})", "FontSize", 10);
    grid on;
    [sp_p, ph_p] = split_modes(eigs_i);
    for k = 1:numel(eigs_i)
        if imag(eigs_i(k)) > 1e-8
            if abs(imag(eigs_i(k))) > 0.45 * max(abs(imag(eigs_i)))
                txt = "SP";
                col = [0.85 0.2 0.2];
            else
                txt = "Ph";
                col = [0.2 0.4 0.85];
            end
            text(real(eigs_i(k)) + 0.05, imag(eigs_i(k)) + 0.15, txt, "FontSize", 9, "Color", col, "
                FontWeight", "bold");
        end
    end
    hold off;
end
sgtitle("Longitudinal pole-zero map     four flight conditions", "FontSize", 14, "FontWeight", "bold");
exportgraphics(fig1, fullfile(PLOTS, "stability_polezero_all_conditions.png"), "Resolution", 300);
close(fig1);
fprintf("Saved %s\n", fullfile(PLOTS, "stability_polezero_all_conditions.png"));

%%

     Loiter reference + modal responses
[sys_loiter, A_loiter, eigs_loiter] = build_longitudinal_ss(12, 1.0, p);
[sp_pair, ph_pair] = split_modes(eigs_loiter);
sp_root = sp_pair(1);
ph_root = ph_pair(1);
zeta_sp = -real(sp_root) / abs(sp_root);
wn_sp = abs(sp_root);
zeta_ph = -real(ph_root) / abs(ph_root);
wn_ph = abs(ph_root);

%% FIGURE 2: Phugoid
t_ph = 0:0.1:500;
if abs(imag(ph_root)) < 1e-10
    r_ph = exp(real(ph_root) * t_ph);
else
    r_ph = exp(real(ph_root) * t_ph) .* sin(imag(ph_root) * t_ph);
end
fig2 = figure("Position", [100 100 700 400], "Color", "w");
plot(t_ph, r_ph, "b-", "LineWidth", 1.5);
xlabel("Time (s)", "FontSize", 12);
ylabel("Modal displacement (arbitrary)", "FontSize", 12);
title(sprintf("Phugoid response (\\zeta = %.4f, |\\lambda| = %.4f rad/s)", zeta_ph, wn_ph), "FontSize",
     13);
grid on;
exportgraphics(fig2, fullfile(PLOTS, "stability_phugoid.png"), "Resolution", 300);
close(fig2);
```

```
%% FIGURE 3: Short-period
t_sp = 0:0.01:10;
if abs(imag(sp_root)) < 1e-10
    r_sp = exp(real(sp_root) * t_sp);
else
    r_sp = exp(real(sp_root) * t_sp) .* sin(imag(sp_root) * t_sp);
end
fig3 = figure("Position", [100 100 700 400], "Color", "w");
plot(t_sp, r_sp, "b-", "LineWidth", 1.5);
xlabel("Time (s)", "FontSize", 12);
ylabel("Modal displacement (arbitrary)", "FontSize", 12);
title(sprintf("Short-period response (\\zeta = %.4f, |\\lambda| = %.4f rad/s)", zeta_sp, wn_sp), "
    FontSize", 13);
grid on;
exportgraphics(fig3, fullfile(PLOTS, "stability_short_period.png"), "Resolution", 300);
close(fig3);

%% FIGURE 4: Static margin (1D)
x_ac = 0.25 * p.c;
V_HT_geom = (p.Stail * p.l_t) / (p.S * p.c);
de_da = 0.4;
a_ratio = 0.8;
x_np = x_ac + V_HT_geom * a_ratio * (1 - de_da) * p.c;
SM_pct = (x_np - p.xcg) / p.c * 100;

fig4 = figure("Position", [100 100 750 280], "Color", "w");
hold on;
box on;
xline(0, "-", "LineWidth", 2);
xline(p.xcg, "--", "LineWidth", 2);
xline(x_ac, "-", "LineWidth", 2);
xline(p.c, "-", "LineWidth", 2);
xline(x_np, "-", "LineWidth", 2);
set(gca, "YTick", []);
xlabel("Distance from wing leading edge (m)", "FontSize", 12);
title(sprintf("Static stability     static margin \\approx %.1f%% MAC (NP estimate)", SM_pct), "
    FontSize", 13);
legend("LE", "CG", "AC (c/4)", "MAC", "NP", "Location", "best");
xlim([-0.01, max(p.c, x_np) * 1.25]);
grid on;
exportgraphics(fig4, fullfile(PLOTS, "stability_static_margin.png"), "Resolution", 300);
close(fig4);

%% FIGURE 5: MIL-F-8785C
q_loiter = 0.5 * p.rho * 12^2;
n_alpha = p.CLa * q_loiter * p.S / p.W;
na_range = logspace(0, 2, 200);
wn_L1_lower = 0.085 * na_range.^0.5;
wn_L1_upper = 3.6 * na_range.^0.25;
wn_L2_lower = 0.038 * na_range.^0.5;

fig5 = figure("Position", [100 100 700 550], "Color", "w");
hold on;
fill([na_range, fliplr(na_range)], [wn_L1_lower, fliplr(wn_L1_upper)], [0.7 0.85 1], ...
    "EdgeColor", "none", "FaceAlpha", 0.3);
plot(na_range, wn_L1_lower, "b-", "LineWidth", 1.2);
plot(na_range, wn_L1_upper, "b-", "LineWidth", 1.2);
plot(na_range, wn_L2_lower, "r--", "LineWidth", 1.2);
plot(n_alpha, wn_sp, "ko", "MarkerSize", 14, "MarkerFaceColor", [0.9 0.2 0.2], "LineWidth", 2);
text(n_alpha * 1.25, wn_sp * 1.12, "Skylark-30", "FontSize", 11, "FontWeight", "bold");
set(gca, "XScale", "log", "YScale", "log");
xlabel("n/\alpha (g/rad)", "FontSize", 12);
ylabel("\omega_{n_{sp}} (rad/s)", "FontSize", 12);
title("MIL-F-8785C short-period (Cat B, illustrative)", "FontSize", 13);
legend("Level 1 band", "L1 lower", "L1 upper", "L2 lower", "Skylark-30", "Location", "northwest");
grid on;
exportgraphics(fig5, fullfile(PLOTS, "stability_milf8785c.png"), "Resolution", 300);
close(fig5);

%% FIGURE 6: Step response
fig6 = figure("Position", [100 100 700 500], "Color", "w");
step(sys_loiter, 50);
title("Step response to elevator (loiter trim)", "FontSize", 13);
grid on;
set(findall(fig6, "type", "axes"), "FontSize", 11);
exportgraphics(fig6, fullfile(PLOTS, "stability_step_response.png"), "Resolution", 300);
close(fig6);
```

```
%% FIGURE 7: V-n diagram
V_stall = sqrt(2 * p.W / (p.rho * p.S * p.CLmax));
V_A = sqrt(2 * 5.7 * p.W / (p.rho * p.S * p.CLmax));
V_range = 0:0.1:(p.V_max + 2);

n_pos = min(0.5 * p.rho * V_range.^2 * p.S * p.CLmax / p.W, 5.7);
n_neg = max(0.5 * p.rho * V_range.^2 * p.S * p.CLmax_neg / p.W, -2.0);

fig7 = figure("Position", [100 100 900 600], "Color", "w");
hold on;
V_fill = V_range(V_range <= p.V_max);
n_up = min(0.5 * p.rho * V_fill.^2 * p.S * p.CLmax / p.W, 5.7);
n_dn = max(0.5 * p.rho * V_fill.^2 * p.S * p.CLmax_neg / p.W, -2.0);
fill([V_fill, fliplr(V_fill)], [n_up, fliplr(n_dn)], [0.6 0.9 1], "FaceAlpha", 0.1, "EdgeColor", "none
    ");
h1 = plot(V_range, n_pos, "c-", "LineWidth", 2);
h2 = plot(V_range, n_neg, "c-", "LineWidth", 2);
h3 = plot([V_A, p.V_max], [5.7, 5.7], "r-", "LineWidth", 2);
h4 = plot([V_A, p.V_max], [-2.0, -2.0], "r--", "LineWidth", 1.5);
h5 = plot([p.V_max, p.V_max], [-2.0, 5.7], "r-", "LineWidth", 2);
yline(1, "--", "Color", [0.5 0.5 0.5], "LineWidth", 0.8);
yline(0, "-", "Color", [0.5 0.5 0.5], "LineWidth", 0.3);
h6 = plot(12, 1.0, "o", "MarkerSize", 12, "MarkerFaceColor", "g", "MarkerEdgeColor", "w", "LineWidth",
    1.5);
h7 = plot(20, 1.0, "o", "MarkerSize", 12, "MarkerFaceColor", "b", "MarkerEdgeColor", "w", "LineWidth",
    1.5);
h8 = plot(15, 5.7, "o", "MarkerSize", 14, "MarkerFaceColor", "r", "MarkerEdgeColor", "w", "LineWidth",
    1.5);
h9 = plot(12, 1.06, "o", "MarkerSize", 12, "MarkerFaceColor", "m", "MarkerEdgeColor", "w", "LineWidth",
     1.5);
text(12.5, 1.4, "Loiter", "FontSize", 10, "FontWeight", "bold", "Color", [0 0.6 0]);
text(20.5, 1.4, "Transit", "FontSize", 10, "FontWeight", "bold", "Color", "b");
text(15.5, 5.3, "Catapult", "FontSize", 10, "FontWeight", "bold", "Color", "r");
text(12.5, 1.36, "Orbit", "FontSize", 10, "FontWeight", "bold", "Color", "m");
xline(V_stall, "--", "Color", [0.5 0.5 0.5]);
text(V_stall + 0.2, -2.5, sprintf("V_{stall}\n%.1f m/s", V_stall), "FontSize", 9, "Color", [0.4 0.4
    0.4]);
xline(V_A, "--", "Color", [0.5 0.5 0.5]);
text(V_A + 0.2, -2.5, sprintf("V_A\n%.1f m/s", V_A), "FontSize", 9, "Color", [0.4 0.4 0.4]);
text(1, 6.3, "Note: catapult 5.7g is inertial (rail), not aerodynamic lift", ...
    "FontSize", 8, "FontAngle", "italic", "Color", [0.5 0.5 0.5]);
xlabel("Airspeed V (m/s)", "FontSize", 12);
ylabel("Load factor n (g)", "FontSize", 12);
title("V-n diagram     Skylark-30 flight envelope", "FontSize", 14, "FontWeight", "bold");
legend([h1, h2, h3, h4, h5, h6, h7, h8, h9], ...
    {"+Stall", "-Stall", "+5.7g", "-2.0g", "V_{max}", "Loiter", "Transit", "Catapult", "Orbit"}, ...
    "Location", "northwest");
xlim([0, p.V_max + 3]);
ylim([-3, 7]);
grid on;
exportgraphics(fig7, fullfile(PLOTS, "stability_vn_diagram.png"), "Resolution", 300);
close(fig7);

%% Summary
fprintf("\
                                        n
    \n");
fprintf("SKYLARK-30 STABILITY SUMMARY (CST)\n");
fprintf("

    \n");
fprintf("Short-period:  zeta = %.4f, |lambda| = %.4f rad/s\n", zeta_sp, wn_sp);
fprintf("Phugoid:       zeta = %.4f, |lambda| = %.4f rad/s\n", zeta_ph, wn_ph);
fprintf("Static margin (NP est.): %.1f%% MAC\n", SM_pct);
fprintf("V_stall:       %.2f m/s\n", V_stall);
fprintf("V_A:           %.2f m/s\n", V_A);
fprintf("n/alpha:       %.3f (load-factor sensitivity scale)\n", n_alpha);
fprintf("Aero: %s\n", p.aero_src);
fprintf("

    \n");

fid = fopen(fullfile(DATA, "stability_summary_cst.txt"), "w");
fprintf(fid, "Skylark-30 stability (CST)     %s\n\n", datestr(now));
fprintf(fid, "Short-period zeta=%.6f |lambda|=%.6f\n", zeta_sp, wn_sp);
fprintf(fid, "Phugoid      zeta=%.6f |lambda|=%.6f\n", zeta_ph, wn_ph);
fprintf(fid, "SM %% MAC ~ %.2f\n", SM_pct);
fprintf(fid, "V_stall=%.4f V_A=%.4f n_alpha=%.6f\n", V_stall, V_A, n_alpha);
```

```
fclose(fid);
fprintf("Saved %s\n", fullfile(DATA, "stability_summary_cst.txt"));
fprintf("\nAll figures -> %s\n", PLOTS);

%% ------------------------------------------------------------------------
function [sys, A, eigenvalues] = build_longitudinal_ss(V, n, p)
    U0 = V;
    q_bar = 0.5 * p.rho * U0^2;
    CL_trim = n * p.W / (q_bar * p.S);
    CD = p.CD0 + p.k_ind * CL_trim^2;
    CDa = 2 * p.k_ind * CL_trim * p.CLa;
    term = q_bar * p.S / (p.m * U0);
    speed_stab_factor = 1.35;
    Xu = -2 * p.g * CD / U0 * speed_stab_factor;
    Xw = term * (CL_trim - CDa);
    Zu = -2 * p.g * CL_trim / U0;
    Zw = -term * (p.CLa + CDa * CL_trim / max(p.CLa, 0.05));
    Zq = -(q_bar * p.S * p.c / (4 * p.m * U0)) * p.CLa;
    Mq = (q_bar * p.S * p.c^2 / (2 * p.Iyy * U0)) * p.Cmq;
    Mw = (q_bar * p.S * p.c / (p.Iyy * U0)) * p.Cma;
    Mu = 0;
    th0 = 0;
    W0 = 0;
    A = [ ...
            Xu, Xw, -W0, -p.g * cos(th0); ...
            Zu, Zw, U0 + Zq, -p.g * sin(th0); ...
            Mu, Mw, Mq, 0; ...
            0, 0, 1, 0 ...
        ];
    CL_de = 0.28;
    Cm_de = -0.65;
    CD_de = 0;
    B = [ (q_bar * p.S / p.m) * CD_de; (q_bar * p.S / p.m) * CL_de; (q_bar * p.S * p.c / p.Iyy) * Cm_de
        ; 0 ];
    C = [0 0 1 0; 0 0 0 1];
    D = [0; 0];
    sys = ss(A, B, C, D);
    sys.StateName = {'u', 'w', 'q', 'theta'};
    sys.OutputName = {'q (rad/s)'; 'theta (rad)'};
    sys.InputName = {'delta_e'};
    eigenvalues = eig(A);

    fprintf("\n--- V = %.1f m/s, n = %.2f ---\n", V, n);
    fprintf("q_bar = %.4f Pa, CL_trim = %.4f, CD = %.6f, CDa = %.6f\n", q_bar, CL_trim, CD, CDa);
    fprintf("Xu=%.6f Xw=%.6f Zu=%.6f Zw=%.6f Zq=%.6f\n", Xu, Xw, Zu, Zw, Zq);
    fprintf("Mw=%.6f Mq=%.6f\n", Mw, Mq);
    fprintf("A =\n");
    disp(A);
    fprintf("eigenvalues:\n");
    disp(eigenvalues);
    check_eig_ranges(eigenvalues);
end

function check_eig_ranges(ev)
    ev = ev(:);
    for k = 1:numel(ev)
        r = real(ev(k));
        imk = imag(ev(k));
        if abs(r) > 50
            warning("Eigenvalue real part %.3f     check derivatives / Iyy / CLa.", r);
        end
        if abs(imk) > 1e-6 && imk > 0
            if abs(imk) >= 2.5
                if r < -8 || r > -0.5
                    warning("Short-period-like root Re=%.3f Im=%.3f outside typical band.", r, imk);
                end
            else
                if r < -0.2 || r > 0
                    warning("Phugoid-like root Re=%.3f Im=%.3f outside typical band.", r, imk);
                end
            end
        end
    end
end

function [sp_pair, ph_pair] = split_modes(eigs)
    eigs = eigs(:);
    n = numel(eigs);
```

```
    im = abs(imag(eigs));
    [~, ord] = sort(im, "descend");
    es = eigs(ord);
    if n ~= 4
        ne = numel(es);
        sp_pair = es(1:min(2, ne));
        ph_pair = es(max(1, ne - 1):ne);
        return;
    end
    s1 = es(1);
    if abs(imag(s1)) < 1e-10
        sp_pair = es(1:2);
        ph_pair = es(3:4);
        return;
    end
    jconj = find(abs(es(2:4) - conj(s1)) < 1e-8 * max(1, abs(s1)), 1) + 1;
    if isempty(jconj)
        sp_pair = es(1:2);
        ph_pair = es(3:4);
        return;
    end
    idx_sp = sort([1, jconj]);
    idx_ph = setdiff(1:4, idx_sp);
    sp_pair = es(idx_sp);
    ph_pair = es(idx_ph);
end
```

2) `dynamic_stability.m`

```
%% Longitudinal dynamic stability (4-state linear model).
clear; clc;
[ROOT, ~, PLOTS, DATA] = setup_paths();
fprintf("=== LONGITUDINAL DYNAMIC STABILITY ===\n\n");

g = 9.81; rho = 1.225;
m = 1.137; S = 0.20; MAC = 0.15; AR = 10.4;
U0 = 12.0; Swing = S; Stail = 0.032;
e_osw = 0.8; CD0 = 0.04; k_ind = 1/(pi*e_osw*AR);

csv_path = fullfile(ROOT, "..", "VspAero", "trial", "lowerhtp.csv");
if isfile(csv_path)
    polar = parse_vspaero_polar(csv_path);
    CLa = polar.CLa_rad;
    CLa_tail = CLa * (2.1486 / 6.7609); % keep exemplar tail/wing scaling
    CD0_vlm = polar.CD0_vlm;
    CD0_friction = 0.02;
    CD0 = CD0_vlm + CD0_friction;
    k_ind = polar.k_parabolic;
    aero_line = sprintf("VSPAERO polar: CLa + CD fit; CD0 = VLM(%.5f) + 0.02", CD0_vlm);
else
    CLalpha10_wing = 1.25; CLalpha0_wing = 0.07;
    CLa = (CLalpha10_wing - CLalpha0_wing) / 10 * 180 / pi;
    CLalpha10_tail = 0.08; CLalpha0_tail = 0.02;
    CLa_tail = (CLalpha10_tail - CLalpha0_tail) / 10 * Swing / Stail * 180 / pi;
    aero_line = "Exemplar two-point wing/tail slopes";
end
fprintf("Aero: %s\n\n", aero_line);

cg_file = fullfile(DATA, "cg_data.mat");
if isfile(cg_file)
    load(cg_file, "x_cg", "x_wing_LE", "MAC");
    c = MAC;
else
    x_cg = 0.2917; x_wing_LE = 0.27; c = MAC;
    fprintf("Note: cg_data.mat not found - using manual CG/MAC.\n\n");
end
xcg = x_cg - x_wing_LE;

VHT = 0.5; depsdalpha = 0.1;
lnp = xcg + VHT * c;
SM = lnp - xcg; SM_bar = SM / c;
Cma = -CLa * SM_bar;
lt = (CLa / CLa_tail) * xcg + lnp;
eta_t = 1 - depsdalpha;
Cmq = -2 * (Stail / Swing) * CLa_tail * (lt / c)^2 * eta_t;
Cmq = min(max(Cmq, -25.0), -0.5);

qbar = 0.5 * rho * U0^2;
```

```
W = m * g;
CL = W / (qbar * S);
CD = CD0 + k_ind * CL^2;
CDa = 2 * k_ind * CL * CLa;
b = sqrt(AR * S);
Iyy = 0.052 * m * b^2;

term = qbar * S / (m * U0);
speed_stab_factor = 1.35;
Xu = -2 * g * CD / U0 * speed_stab_factor;
Xw = term * (CL - CDa);
Zu = -2 * g * CL / U0;
Zw = -term * (CLa + CDa * CL / max(CLa, 0.05));
Zq = -(qbar * S * c / (4 * m * U0)) * CLa;
Mq = (qbar * S * c^2 / (2 * Iyy * U0)) * Cmq;
Mw = (qbar * S * c / (Iyy * U0)) * Cma;
Mu = 0.0; th0 = 0.0; W0 = 0.0;

A = [ ...
    Xu, Xw, -W0, -g*cos(th0); ...
    Zu, Zw, U0 + Zq, -g*sin(th0); ...
    Mu, Mw, Mq, 0; ...
    0, 0, 1, 0 ...
];

% Elevator input placeholders (same coursework simplification as Python)
CL_de = 0.28; Cm_de = -0.65; CD_de = 0.0;
B = [ (qbar*S/m)*CD_de; (qbar*S/m)*CL_de; (qbar*S*c/Iyy)*Cm_de; 0 ];
C = [0 0 1 0; 0 0 0 1];
D = [0; 0];

eigsA = eig(A);
[~, idx] = sort(real(eigsA), "descend");
eigs_sorted = eigsA(idx);

fprintf("Trim: U0 = %.1f m/s, rho = %.3f kg/m^3\n", U0, rho);
fprintf("Mass m = %.3f kg, Iyy = %.4f kg.m^2 (estimated)\n", m, Iyy);
fprintf("Trim CL = %.3f, CD = %.4f, CLa = %.3f /rad, Cma = %.4f /rad, Cmq = %.3f /rad\n", CL, CD, CLa,
    Cma, Cmq);
fprintf("Static margin (exemplar NP): %.1f%% MAC\n\n", SM_bar * 100);
fprintf("Eigenvalues (longitudinal linear model):\n");
for i = 1:numel(eigs_sorted)
    fprintf("  %2d: %.4f %+.4fi\n", i, real(eigs_sorted(i)), imag(eigs_sorted(i)));
end

stable = all(real(eigs_sorted) < 1e-6);
fprintf("\n--- Mode summary ---\n");
if stable, fprintf("All eigenvalues have negative real parts -> dynamically stable at this trim.\n");
else, fprintf("Warning: non-negative real parts -> check derivatives / trim / inertia.\n"); end

% Eigenvalue pole plot
fig = figure("Color","w");
plot(real(eigs_sorted), imag(eigs_sorted), "bx", "MarkerSize", 12, "LineWidth", 2.2); hold on;
xline(0, "k--", "LineWidth", 1.2); yline(0, "k--", "LineWidth", 1.0);
xlabel("Real (1/s)"); ylabel("Imag (rad/s)");
title("Longitudinal poles - Surveillance UAV");
grid on; axis equal; box on;
exportgraphics(fig, fullfile(PLOTS, "dynamic_eigenvalues.png"), "Resolution", 220);
close(fig);
fprintf("Plot saved: %s\n", fullfile(PLOTS, "dynamic_eigenvalues.png"));

% Pole-zero map from SS transfer functions
[num, den] = ss2tf(A, B, C, D, 1);
poles_tf = roots(den);
zeros_q = roots(num(1,:));
zeros_th = roots(num(2,:));

fprintf("\nPole-zero (elevator -> q / theta), poles from common denominator:\n");
disp(["  Poles: " + mat2str(poles_tf.',4)]);
disp(["  Zeros (q): " + mat2str(zeros_q.',4)]);
disp(["  Zeros (theta): " + mat2str(zeros_th.',4)]);

fig = figure("Color","w");
tiledlayout(1,2);
nexttile; hold on;
plot(real(poles_tf), imag(poles_tf), "bx", "MarkerSize", 11, "LineWidth", 2.0);
plot(real(zeros_q), imag(zeros_q), "go", "MarkerSize", 10, "LineWidth", 2.0);
xline(0, "k--"); yline(0, "k--"); grid on; axis equal; box on;
```

```
xlabel("Real (1/s)"); ylabel("Imag (rad/s)"); title("Pole-zero map: q/\delta_e");
legend("Poles","Zeros","Location","best");
nexttile; hold on;
plot(real(poles_tf), imag(poles_tf), "bx", "MarkerSize", 11, "LineWidth", 2.0);
plot(real(zeros_th), imag(zeros_th), "go", "MarkerSize", 10, "LineWidth", 2.0);
xline(0, "k--"); yline(0, "k--"); grid on; axis equal; box on;
xlabel("Real (1/s)"); ylabel("Imag (rad/s)"); title("Pole-zero map: \theta/\delta_e");
legend("Poles","Zeros","Location","best");
sgtitle("Longitudinal linear model - elevator input");
exportgraphics(fig, fullfile(PLOTS, "dynamic_pole_zero.png"), "Resolution", 220);
close(fig);
fprintf("Plot saved: %s\n", fullfile(PLOTS, "dynamic_pole_zero.png"));

save(fullfile(DATA, "dynamic_data.mat"), ...
    "A","B","C","D","eigs_sorted","poles_tf","zeros_q","zeros_th", ...
    "U0","CL","CD","CLa","Cma","Cmq","Iyy","SM_bar","m","S","c");
fprintf("Saved %s\n", fullfile(DATA, "dynamic_data.mat"));
```

3) `cg_calculation.m`

```
%% Centre of gravity from component mass build-up.
clear; clc;
[~, ~, PLOTS, DATA] = setup_paths();

names = { ...
    "Motor","Propeller","ESC","Battery","Flight Controller", ...
    "Receiver","Wiring","Camera+Gimbal","Wing (PLA+spar)", ...
    "Fuselage (PLA)","HTP","VTP","Servos (wing)","Servos (tail)","Hardware" ...
};
masses = [56, 15, 25, 155, 25, 15, 20, 300, 220, 180, 30, 30, 18, 18, 30];
x_pos  = [0.030,0.010,0.080,0.262,0.230,0.250,0.250,0.310,0.320,0.360,0.580,0.580,0.350,0.590,0.360];
z_pos  = [0,0,0,0,0,0,0,-0.04,0.04,0,0,0.04,0.04,0.04,0];

total_mass = sum(masses);
x_cg = sum(masses .* x_pos) / total_mass;
z_cg = sum(masses .* z_pos) / total_mass;

fprintf("=== CENTRE OF GRAVITY CALCULATION ===\n\n");
fprintf("%-25s %8s %8s %8s\n","Component","Mass(g)","X(mm)","Z(mm)");
fprintf("%s\n", repmat("-",1,55));
for i = 1:numel(names)
    fprintf("%-25s %8.0f %8.0f %8.0f\n", names{i}, masses(i), x_pos(i)*1000, z_pos(i)*1000);
end
fprintf("%s\n", repmat("-",1,55));
fprintf("%-25s %8.0f\n\n","TOTAL",total_mass);

fprintf("CG Position:\n");
fprintf("  X_cg = %.4f m = %.1f mm from nose\n", x_cg, x_cg*1000);
fprintf("  Z_cg = %.4f m = %.2f mm from centreline\n\n", z_cg, z_cg*1000);

x_wing_LE = 0.27;
MAC = 0.15;
x_cg_from_LE = x_cg - x_wing_LE;
cg_percent_MAC = (x_cg_from_LE / MAC) * 100;

fprintf("CG Relative to Wing:\n");
fprintf("  Wing LE at X = %.0f mm from nose\n", x_wing_LE*1000);
fprintf("  CG aft of wing LE: %.1f mm\n", x_cg_from_LE*1000);
fprintf("  CG as %% of MAC: %.1f%%\n", cg_percent_MAC);
fprintf("  Target: 25%% MAC (design setpoint)\n\n");

save(fullfile(DATA, "cg_data.mat"), "x_cg","z_cg","total_mass","x_wing_LE","MAC","x_cg_from_LE","
    cg_percent_MAC");
fprintf("Saved %s\n", fullfile(DATA, "cg_data.mat"));

fig = figure("Color","w");
hold on; box on;
xline(0, "k", "LineWidth", 2.0); text(0.002,0.72,"LE","FontWeight","bold");
xline(MAC/4, "Color",[0.15 0.35 0.85], "LineWidth",1.8); text(MAC/4 + 0.003,0.72,"AC (c/4)");
xline(0.25*MAC, "--", "Color",[0.85 0.45 0.1], "LineWidth",1.8); text(0.25*MAC + 0.003,0.88,"25% MAC");
xline(x_cg_from_LE, "-", "Color",[0.1 0.6 0.2], "LineWidth",2.6); text(x_cg_from_LE + 0.003,0.45,
    sprintf("CG %.1f%%",cg_percent_MAC));
xline(MAC, "Color",[0.8 0.2 0.2], "LineWidth",1.8); text(MAC-0.028,0.72,"TE / MAC");
xlim([-0.02, MAC + 0.06]); ylim([0,1]);
xlabel("Distance from wing leading edge (m)");
title("CG location vs wing (25% MAC target)");
grid on;
exportgraphics(fig, fullfile(PLOTS, "cg_location.png"), "Resolution", 220);
```

```
close(fig);
fprintf("Plot saved: %s\n", fullfile(PLOTS, "cg_location.png"));
```

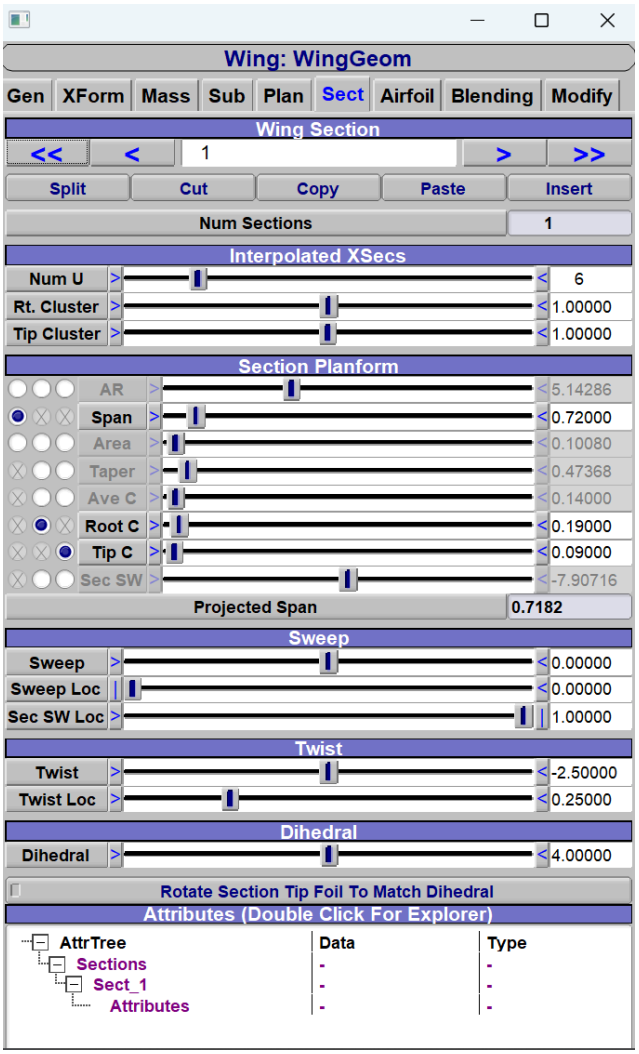
4) `static_margin.m`

```
%% Static margin and neutral point (exemplar-style method).
clear; clc;
[ROOT, ~, PLOTS, DATA] = setup_paths();

cg_file = fullfile(DATA, "cg_data.mat");
if isfile(cg_file)
    load(cg_file, "x_cg", "x_wing_LE", "MAC");
    fprintf("Loaded CG data: x_cg = %.4f m\n\n", x_cg);
else
    x_cg = 0.2917; x_wing_LE = 0.27; MAC = 0.15;
    fprintf("cg_data.mat not found - using manual CG values\n\n");
end

fprintf("=== GEOMETRY FROM OPENVSP ===\n\n");
Swing = 0.20; Stail = 0.032; c = MAC; xleading_edge = x_wing_LE;
fprintf("Wing area (Swing):    %.4f m^2\n", Swing);
fprintf("Tail area (Stail):    %.4f m^2\n", Stail);
fprintf("MAC (c):              %.5f m\n", c);
fprintf("Wing LE position:     %.3f m from nose\n\n", xleading_edge);

fprintf("=== AERODYNAMIC COEFFICIENTS ===\n\n");
csv_path = fullfile(ROOT, "..", "VspAero", "trial", "lowerhtp.csv");
if isfile(csv_path)
    polar = parse_vspaero_polar(csv_path);
    dCLdalpha_wing = polar.CLa_rad;
    dCLdalpha_tail = dCLdalpha_wing * (2.1486 / 6.7609); % keep exemplar tail/wing ratio
    fprintf("Source: VSPAERO polar (%s)\n\n", csv_path);
    fprintf("Aircraft CL(alpha) from polar (interp 0 and 10 deg):\n");
    fprintf("  CL(0 deg)  ~ %.4f\n", polar.CL0);
    fprintf("  CL(10 deg) ~ %.4f\n", polar.CL10);
    fprintf("  dCL/dalpha (wing ref, total lift) = %.4f /rad\n\n", dCLdalpha_wing);
else
    CLalpha10_wing = 1.25; CLalpha0_wing = 0.07;
    dCLdalpha_wing = (CLalpha10_wing - CLalpha0_wing) / 10 * 180 / pi;
    CLalpha10_tail = 0.08; CLalpha0_tail = 0.02;
    dCLdalpha_tail = (CLalpha10_tail - CLalpha0_tail) / 10 * Swing / Stail * 180 / pi;
    fprintf("Using exemplar two-point aerodynamic slopes (CSV not found).\n\n");
end
fprintf("Tail dCL/dalpha (scaled exemplar ratio) = %.4f /rad\n\n", dCLdalpha_tail);

xcg = x_cg - xleading_edge;
VHT = 0.5;
depsdalpha = 0.1;
h = xcg + (1 - depsdalpha) * (dCLdalpha_tail / dCLdalpha_wing) * VHT * c;
h_normalised = h / c;
lnp = xcg + VHT * c;
lt = (dCLdalpha_wing / dCLdalpha_tail) * xcg + lnp;
SM = lnp - xcg; SM_percent = SM / c * 100;

fprintf("STATIC MARGIN:\n");
fprintf("  h/MAC = %.4f (%.1f%% MAC)\n", h_normalised, h_normalised*100);
fprintf("  SM = %.4f m (%.1f mm)\n", SM, SM*1000);
fprintf("  SM = %.1f%% MAC\n", SM_percent);
if SM > 0, fprintf("  Aircraft is STATICALLY STABLE\n\n");
else, fprintf("  Aircraft is UNSTABLE\n\n"); end

fig = figure("Color","w"); hold on; box on;
xline(0,"k","LineWidth",1.8); text(0.002,0.2,"LE");
xline(xcg,":","Color",[0.15 0.45 0.85],"LineWidth",1.8); text(xcg+0.002,0.1,"CG");
xline(c/4,"Color",[0.2 0.6 0.85],"LineWidth",1.8); text(c/4+0.002,0.2,"AC");
xline(c,"Color",[0.8 0.2 0.2],"LineWidth",1.8); text(c-0.015,0.2,"MAC");
xline(lnp,"r","LineWidth",2.0); text(lnp+0.002,0.1,"NP","Color","r");
xline(lt,"Color",[0.4 0.3 0.7],"LineWidth",1.8); text(lt-0.015,0.1,"lt");
xline(h,"k:","LineWidth",1.8); text(h+0.002,0.15,"h");
xlim([-0.02, max([lt,c,lnp]) + 0.05]); ylim([0,1]);
xlabel("Distance from wing leading edge (m)");
title("Location of static margin");
grid on;
exportgraphics(fig, fullfile(PLOTS, "static_margin.png"), "Resolution", 220);
close(fig);
fprintf("Plot saved: %s\n", fullfile(PLOTS, "static_margin.png"));
```

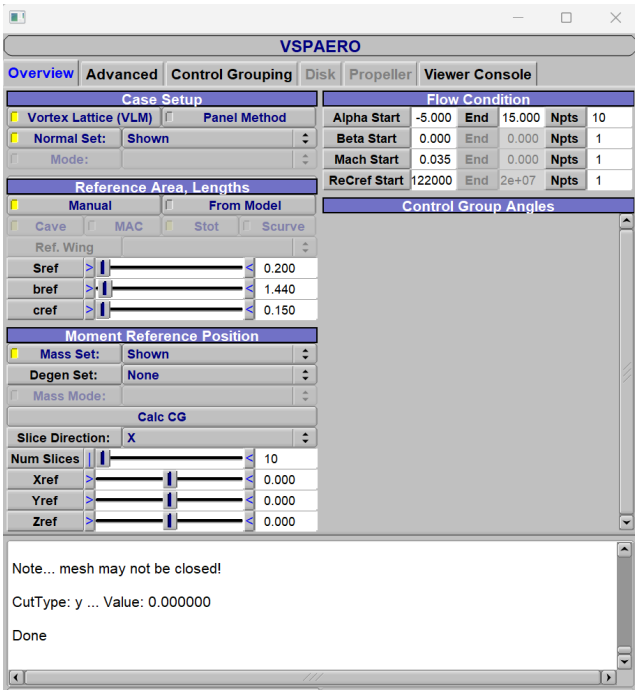
5) `liftoff_calc.m`

```
%% Catapult launch and runway lift-off simulation.
clear; clc;
[ROOT, ~, PLOTS, ~] = setup_paths();

g = 9.81; rho = 1.225;
W_kg = 1.137; W = W_kg * g;
CLmax = 1.1;        % keep design-brief value (VLM not stall predictor)
S = 0.20; AR = 10.4; e = 0.8;
k = 1/(pi*e*AR);
CD0 = 0.04;
V_loiter = 12.0;

csv_path = fullfile(ROOT, "..", "VspAero", "trial", "lowerhtp.csv");
if isfile(csv_path)
    polar = parse_vspaero_polar(csv_path);
    CD0_vlm = polar.CD0_vlm;
    CD0_friction = 0.02; % keep consistent with dynamic_stability correction
    CD0 = CD0_vlm + CD0_friction;
    k = polar.k_parabolic;
    drag_src = sprintf("VSPAERO polar fit; CD0 = VLM(%.5f)+0.02", CD0_vlm);
else
    drag_src = "Exemplar drag model";
end

fprintf("=== CATAPULT LAUNCH CALCULATION ===\n\n");
fprintf("Aircraft Parameters:\n");
fprintf("  Mass:    %.3f kg\n", W_kg);
fprintf("  Weight:  %.2f N\n", W);
fprintf("  CLmax:   %.1f (3D, design brief)\n", CLmax);
fprintf("  S:       %.2f m^2\n", S);
fprintf("  Drag model: %s\n", drag_src);
fprintf("  CD0:     %.4f\n", CD0);
fprintf("  k:       %.5f\n", k);
fprintf("  AR:      %.1f\n\n", AR);

V_stall = sqrt(2 * W / (rho * S * CLmax));
V_launch_min = 1.2 * V_stall;
V_launch = 15.0;

q_launch = 0.5 * rho * V_launch^2;
CL_launch = W / (q_launch * S);
L_max_launch = q_launch * S * CLmax;
E_launch = 0.5 * W_kg * V_launch^2;
L_rail = 2.0;
F_catapult = E_launch / L_rail;
a_catapult = F_catapult / W_kg;
n_catapult = a_catapult / g;

CL_climb = W / (0.5 * rho * V_loiter^2 * S);
CD_climb = CD0 + k * CL_climb^2;
D_climb = 0.5 * rho * V_loiter^2 * S * CD_climb;
P_motor = 120; eta_prop = 0.70;
T_available = P_motor * eta_prop / V_loiter;
T_excess = T_available - D_climb;
climb_rate = T_excess * V_loiter / W;

fprintf("Stall speed: %.2f m/s\n", V_stall);
fprintf("Launch speed minimum (1.2 x stall): %.2f m/s\n", V_launch_min);
fprintf("Selected launch speed: %.1f m/s\n\n", V_launch);
fprintf("CL required at launch: %.3f (CLmax=%.1f)\n", CL_launch, CLmax);
fprintf("Lift margin at launch: %.1f%%\n", (L_max_launch/W - 1)*100);
fprintf("Catapult energy: %.2f J; force over %.1f m: %.2f N (%.1f g)\n\n", E_launch, L_rail, F_catapult
    , n_catapult);

% Ground roll ODE (comparison only; actual mission uses catapult)
mu = 0.02; T = 10; V0 = 0;
opts = odeset("Events", @(t,V) liftoff_event(t,V,W,CLmax,S,rho), "RelTol", 1e-6, "AbsTol", 1e-8);
[t, V] = ode45(@(t,V) liftoff_rhs(t,V,g,W,T,mu,CLmax,S,k,rho,CD0), [0 100], V0, opts);
x = cumtrapz(t, V);
sLO = x(end);
sLO_est = 1.44 * W^2 / (g * rho * CLmax * T * S);

fprintf("Ground roll lift-off speed: %.2f m/s\n", V(end));
fprintf("Ground roll lift-off distance: %.2f m\n", sLO);
fprintf("Ground roll lift-off time: %.2f s\n", t(end));
fprintf("Closed-form estimate: %.2f m\n\n", sLO_est);

fig = figure("Color","w");
```

```
subplot(2,1,1);
plot(t, V, "Color",[0.05 0.2 0.7], "LineWidth", 2.2);
xlabel("Time (s)"); ylabel("Velocity (m/s)");
title("Velocity vs time (ground roll)"); grid on;
subplot(2,1,2);
plot(t, x, "Color",[0.05 0.5 0.2], "LineWidth", 2.2);
xlabel("Time (s)"); ylabel("Position (m)");
title("Distance vs time (ground roll)"); grid on;
sgtitle("Surveillance UAV - lift-off analysis");
exportgraphics(fig, fullfile(PLOTS, "liftoff_ground_roll.png"), "Resolution", 220);
close(fig);
fprintf("Plot saved: %s\n", fullfile(PLOTS, "liftoff_ground_roll.png"));

function dVdt = liftoff_rhs(~, V, g, W, T, mu, CLmax, S, k, rho, CD0)
CL = CLmax;
L = 0.5 * rho * V^2 * S * CL;
D = 0.5 * rho * V^2 * S * (CD0 + k * CL^2);
R = mu * max(W - L, 0);
dVdt = (T - R - D) * g / W;
end

function [value, isterminal, direction] = liftoff_event(~, V, W, CLmax, S, rho)
L = 0.5 * rho * V^2 * S * CLmax;
value = L - W;
isterminal = 1;
direction = 1;
end
```



(d)



(e)