

Final Project

B-21 Penetrating-Strike Route Planning Under Integrated Threat

Individual · 20% of the final grade, in lieu of a final exam · issued L31 · final paper
due at the start of L41

Background

Every block of this course built one piece of a mission plan. Block 1 gave you the radar range law, $R \propto \sigma^{1/4}$, and the layered air defense in which an early-warning radar cues the engagement radars (L7). Block 2 taught you to listen and to locate: a bearing is a line of position, and the geometry of the bearings sets the size of the fix (L13–L14); a jammer wins or loses against the radar’s echo through J/S and burn-through (L15, L17). Block 3 showed that the B-21 signature is a *pattern*, not a number: its radar cross section and infrared intensity change with the aspect it presents (L25–L28, Project 3). Block 4 turns those pieces into survivability: the kill chain as a survival metric (L31), threat rings and engagement zones (L32), the information geometry of a passive network (L33), the route itself (L34), tracking a threat you did not expect (L35), the support package (L36), and the trends that limit all of it (L37).

This project puts the pieces on one map. A B-21 must penetrate an integrated air defense, strike a fixed target, and come home on a fixed fuel load. You are the mission planner. You will build the threat picture, turn the kill chain into a cost that a shortest-path algorithm can minimize, plan the route under the fuel limit, measure how well the adversary’s passive network can track you along it, decide what a stand-off jammer and one SEAD nomination buy, react to a pop-up threat, and recommend the route to fly with its residual risk. You will write it up as a short journal article in IEEE format, tie its claims to the course lessons that support them, and review two of your classmates’ drafts. An issued MATLAB toolkit carries the models, so your time goes to the decisions and to the paper.

Scenario

The map. A notional defended region 400 km (East) by 300 km (North), with positions given as North–East coordinates in km from its south-west corner. Bearings are measured clockwise from North, $\beta = \text{atan2}(\Delta E, \Delta N)$, as in Project 2 (this handout keeps θ for aspect). The B-21 enters at the ingress point $A = (290, 0)$ on the west edge, strikes the target $T = (140, 330)$, and leaves through the egress point $B = (30, 0)$, also on the west edge. Ingress ($A \rightarrow T$) and egress ($T \rightarrow B$) are separate legs; they may differ, and the egress leg flies away from the defenses around the target.

The aircraft. Constant speed $v_t = 0.25$ km/s at a constant altitude of 12 km. At that altitude the radar horizon, $d = 4.12(\sqrt{h_r} + \sqrt{h_t})$ km with heights in meters, exceeds 450 km for any site, so the horizon never binds and detection is set by the range law alone. The signature is the Project 3 pattern, `B21_SignaturePattern.csv`: the X-band column for the SAM radars, the VHF column for the early-warning radar, and the IR column for the infrared threat. **Fuel limit:** the ingress and egress path lengths together may not exceed $L_{\max} = 850$ km.

The integrated air defense (Table 1). A VHF early-warning radar (EWR) detects and cues but carries no weapon. One long-range SAM (L1) and three medium-range SAMs (M1–M3) engage

with X-band radars. A short-range infrared point defense (SH, the MANPADS/SHORAD class of Project 3) sits at the target. Each weapon site is described by its detection reference, its missile kinematics, and its kill chain (Table 2).

Table 1: Notional threat laydown (`FP_ThreatLaydown.csv`). The reference is the signature the site just detects at its reference range (σ_{ref} at R_{ref} , or J_{ref} at R_{ref} for SH). The MEZ radius is $(v_m + v_t) t_{\text{max}}$; the EWR carries no weapon.

Site	Class	Band	(N, E) [km]	Reference	v_m [km/s]	t_{max} [s]	MEZ [km]
EWR	early warning	VHF	(180, 225)	2 m ² @ 90 km	—	—	—
L1	long-range SAM	X	(195, 310)	0.03 m ² @ 70 km	1.35	50	80
M1	medium-range SAM	X	(250, 150)	0.03 m ² @ 45 km	1.00	32	40
M2	medium-range SAM	X	(130, 275)	0.03 m ² @ 45 km	1.00	32	40
M3	medium-range SAM	X	(128, 316)	0.03 m ² @ 45 km	1.00	32	40
SH	IR point defense	IR	(140, 330)	300 W/sr @ 8 km	0.55	7.5	6

Table 2: Kill-chain links and radar data for each weapon site. $P_{T|D}$ takes its cued value where the EWR holds the B-21, its autonomous value elsewhere (SH has no cueing). Cycle is the shot cycle; $P_t G_t$, the sidelobe level (SLL) and the beamwidth (BW) are the victim-radar data for stand-off jamming. The 8° beamwidth is a notional teaching value, wide on purpose so that a jamming strobe is wide enough to fly on the 5 km grid.

Site	$P_{T D}$ (cued/aut.)	$P_{I T}$	$P_{H I}$	$P_{K H}$	Cycle [s]	$P_t G_t$ [dBW]	SLL [dB]	BW [deg]
L1	0.7 / 0.4	0.9	0.4	0.7	60	94	−40	8
M1, M2	0.7 / 0.4	0.9	0.4	0.6	40	85	−40	8
M3	0.7 / 0.4	0.9	0.4	0.6	40	85	−35	8
SH	0.8 / 0.8	0.9	0.4	0.25	30	—	—	—

The passive network. Three bearing-only infrared search-and-track (IRST) sites watch the B-21 without radiating (Table 3). Each reports a bearing with noise σ_β (column `sigma_theta_deg`; σ_θ in L14 and L33) whenever the B-21 is within its maximum range.

Table 3: Adversary passive network (rows P1–P3 of `FP_ThreatLaydown.csv`).

Site	(N, E) [km]	σ_β [deg]	Maximum range [km]
P1	(205, 60)	0.5	350
P2	(205, 330)	0.5	350
P3	(95, 360)	0.5	120

The pop-up. Intelligence reports a mobile short-range SAM (PU) near (210, 170), uncertain by ± 10 km (1σ in each axis). It is silent at planning time, so plan T3 without it. It radiates the first time the B-21 comes within 100 km of its reported position; from then on your radar warning receiver (RWR) measures its bearing with $\sigma_\beta = 0.5^\circ$, one independent bearing per 4 km of track (the Project 2 budget). Its reported position is your prior: plan the CRLB at that point. Its parameters, in the PU row of the CSV, are: X band, 0.03 m² at 30 km; $v_m = 0.75$ km/s, $t_{\text{max}} = 25$ s (MEZ 25 km); $P_{T|D} = 0.7/0.4$, $P_{I|T} = 0.9$, $P_{H|I} = 0.5$, $P_{K|H} = 0.5$; shot cycle 20 s.

Support (Table 4). One X-band **stand-off jammer** (SOJ), flying one of the five candidate

orbits in `FP_OrbitCandidates.csv` (Table 5). Its antenna can hold its mainlobe on only *one* victim radar at a time, which must be L1, M1, M2 or M3 (the pop-up cannot be jammed); treat the other radars as unjammed. You also get **one SEAD nomination**: one radar-guided SAM site (L1, M1, M2 or M3) is suppressed for the strike window. SEAD cannot nominate the IR point defense (the IR floor stays) or the EWR.

Table 4: Support assets (values in `FP_Scenario.csv`).

Asset	Parameters	Rule
SOJ	X band; $P_j G_j = 25$ dBW toward the victim; $B_j/B_r = 10$ dB; burn-through at $J/S = 10$ dB	orbit (a fixed point): one of the candidates O1–O5; mainlobe on one victim radar (L1, M1, M2 or M3)
SEAD	hazard factor 0.25	one radar-guided SAM (L1, M1–M3), not SH or the EWR; its λ is multiplied by the factor

Table 5: Candidate SOJ orbits (`FP_OrbitCandidates.csv`). Every one is legal: on the map and at least 20 km outside every SAM’s MEZ, the pop-up’s prior included (`FPToolkit.orbitLegal` checks it).

Orbit	O1	O2	O3	O4	O5
(N, E) [km]	(68, 284)	(258, 390)	(119, 376)	(290, 250)	(300, 190)

Models

These are *teaching models*: simple enough to compute on a laptop, detailed enough to make the trades real. State them in your paper as models, and say where they break.

Aspect. Aspect is measured from the aircraft’s nose: 0° nose-on, 180° tail-on. A B-21 on heading ψ at bearing ϕ from a radar (so the radar lies at bearing $\phi + 180^\circ$ from the aircraft) presents the aspect

$$\theta = \text{wrap}_{360}(\phi + 180^\circ - \psi).$$

Detection contour. A radar detects the B-21 out to

$$R_{\text{det}}(\theta) = R_{\text{ref}} \left(\frac{\sigma_b(\theta)}{\sigma_{\text{ref}}} \right)^{1/4},$$

with $\sigma_b(\theta)$ read from the pattern in the radar’s band (the table is in dBsm; interpolate in dB and convert with $\sigma = 10^{\text{dBsm}/10}$). The IR point defense acquires out to $R = R_{\text{ref}} \sqrt{J(\theta)/J_{\text{ref}}}$, capped at its lethal range. Because the pattern changes with aspect, the contour a site draws depends on *your* heading.

Engagement zones. With a constant average missile speed v_m and maximum time of flight t_{max} , the **MEZ** (the outer launch boundary against a target flying at the site) is $R_{\text{MEZ}} = (v_m + v_t) t_{\text{max}}$, and the **LEZ** (the no-escape zone: the missile catches even a target turning directly away) is $R_{\text{LEZ}} = (v_m - v_t) t_{\text{max}}$. Use the circular MEZ for the lethal zone; draw the LEZ on your threat picture.

Lethal zone and hazard. A site can engage you where you are inside *both* its detection contour and its MEZ. While you are inside, it fires once every Δs km of track ($\Delta s = \text{shot cycle} \times v_t$), and each shot kills with $p_{\text{shot}} = P_{T|D} \cdot P_{I|T} \cdot P_{H|I} \cdot P_{K|H}$. The survival over ℓ km of exposure is $(1 - p_{\text{shot}})^{\ell/\Delta s} = e^{-\lambda\ell}$, with the **hazard**

$$\lambda = \frac{-\ln(1 - p_{\text{shot}})}{\Delta s} \quad [\text{per km}].$$

Cueing: a radar SAM uses its cued $P_{T|D}$ at any point where the EWR's own detection contour (VHF, aspect-dependent) holds the B-21, and its autonomous value elsewhere. For a route, the risks add over sites and over the route:

$$r = \sum_i \int \lambda_i ds = \sum_i \lambda_i \ell_i, \quad P_S = e^{-r}.$$

Compute P_S for the ingress and the egress legs separately; the route's P_S is their product. The **binding threat** on a leg is the site that contributes the largest share of that leg's risk.

Route planning. Lay an 8-connected grid over the map at the spacing in `FP_Scenario.csv` (5 km; the points A , T and B are nodes). Plan on this issued grid; a finer or coarser grid is a sensitivity to report, not your baseline. Each edge e has a length ℓ_e and a risk $r_e = \sum_i \lambda_i \ell_{i,e}$; find $\ell_{i,e}$ by sampling points along the edge (every 0.5 km or finer), each with the edge's heading (the toolkit uses 15 points per edge). Edge cost $c_e = \ell_e + (w + 10^{-6})r_e + 10^{-12}d_e\ell_e$, with $w = w_r/w_d$ in km per unit of risk. The two small terms settle exact ties, and here they matter: much of the map is risk-free, so many routes tie exactly in length and risk. The 10^{-6} (the L34 rule) sends exact length ties to the safer route. In the last term, d_e is the distance (km) from the edge's midpoint to the infinite line through the leg's two endpoints (A – T for ingress, T – B for egress, the re-plan's start to T in T6); it sends any remaining tie to the route nearest that line. Use exactly this rule (`FPToolkit.route` does), so that for a given w and threat set your route is unique and the same as your classmates'. Because an edge's heading sets the aspect every radar sees, $a \rightarrow b$ and $b \rightarrow a$ cost differently: the graph is directed (`MATLAB digraph, shortestpath`). Sweeping w on a log grid from 0 until the route stops changing ($w \geq 10^4$ km here) traces each leg's risk-versus-length **Pareto front**. The fuel limit is applied to the sweep, not inside Dijkstra: fly the lowest-risk ingress–egress pair whose total length is at most L_{max} . An **aspect-blind** plan uses the nose-on signature everywhere, which makes every edge cost the same both ways (`graph`). Because each edge is scored at its own heading, a low-risk route can zig-zag between edge directions where the aspect pattern rewards it, most visibly near the target: that is an artifact of per-edge aspect scoring on a grid, not a bug, and a natural limitation to discuss.

Information geometry. A bearing-only sensor k at range R_k from a point contributes the row $\mathbf{h}_k = [-\Delta E_k, \Delta N_k]/R_k^2$ (with $\Delta N_k, \Delta E_k$ measured from the sensor to the point) to the Fisher information

$$\mathbf{J} = \sum_k \frac{\mathbf{h}_k^\top \mathbf{h}_k}{\sigma_\beta^2} \quad (\sigma_\beta \text{ in radians}), \quad \text{Cov}(\hat{\mathbf{p}}) \succeq \mathbf{J}^{-1}, \quad \text{DRMS} = \sqrt{\text{tr } \mathbf{J}^{-1}}.$$

The same formula serves in both directions: their IRST network locating you (T4, sensors within their maximum range), and your RWR locating the pop-up (T6, the sum running over your bearings along the leg, evaluated at the pop-up's prior position). A **seam** is where $\mathbf{J}$ is ill-conditioned: too few sensors hold you, or the ones that do look along nearly the same line. Where fewer than two

sensors hold you, or $\mathbf{J}$ is singular, the DRMS is infinite: plot it at a stated cap (for example 100 km) and mark those stretches.

Stand-off jamming. For a target at range R_t from the victim radar and the jammer at range R_j , with the radar's receive gain toward the jammer G'_r relative to its mainbeam gain G_r ,

$$\frac{J}{S} = \frac{4\pi P_j G_j R_t^4}{P_t G_t \sigma R_j^2} \cdot \frac{G'_r}{G_r} \cdot \frac{B_r}{B_j}, \quad R_{bt} = \left[\frac{10 P_t G_t \sigma R_j^2 B_j}{4\pi P_j G_j (G'_r/G_r) B_r} \right]^{1/4},$$

with burn-through at $J/S = 10$ (10 dB), as in L36. Work in SI units: P in W, gains and B_j/B_r as linear ratios, σ in m^2 , R_t and R_j in m (with km the R_{bt} formula is off by $\sqrt{1000}$). $G'_r/G_r = 1$ when the jammer's azimuth from the radar is within half a beamwidth of the B-21's azimuth (the jamming **strobe**), and equals the sidelobe level otherwise. The jammed detection contour is $R(\phi) = \min(R_{det}(\phi), R_{bt}(\phi))$, with $\sigma = \sigma_b(\theta)$ at the aspect presented. Inside the strobe the radar holds you only within the short mainlobe burn-through range, so the strobe is a corridor you can fly; elsewhere the jammer enters only through the sidelobes.

Tools

Scenario files. `FP_ThreatLaydown.csv` holds one row per adversary site (EWR, L1, M1–M3, SH, the pop-up PU, and the IRST sites P1–P3) with every value in Tables 1–3; a blank means the column does not apply to that site. `FP_Scenario.csv` holds the scalars as `name,value,unit,description` rows: the map, A , T , B , speed, altitude, $L_{\max}$, the grid spacing, the SOJ and SEAD values, and the pop-up and RWR values. `FP_OrbitCandidates.csv` lists the five candidate SOJ orbits (Table 5). `B21_SignaturePattern.csv` is the Project 3 pattern (`az_deg`, `RCS_VHF_dBsm`, `RCS_X_dBsm`, `IR_Wsr`, 0–355° every 5°).

The toolkit. `FPToolkit.m` implements every model in this handout as a static method (Table 6); call it as `FPToolkit.route(...)`. It computes whatever you ask for and makes no decisions: it never picks the weight w , the ingress–egress pair, the support plan or the maneuver — those, and defending them, are the project. `help FPToolkit` lists the methods, and `help FPToolkit.method` documents each one's inputs and outputs with an example. `score` samples any polyline with the same rule `edgeRisk` uses on a grid edge, so on a route from `route` it returns exactly that route's risk; use `route`'s own `rt.per` for grid routes and `score` for legs off the grid. You are expected to use it, and still to understand every model it implements: the Models section above is what your paper states, T2 is a hand calculation that `FPToolkit.chain` only checks, and you must be able to explain any number the toolkit returns to you.

Starter script. `L38_FinalProjectStarter.m` is a short tour of the toolkit: it loads the scenario, draws the laydown with every site's MEZ and one site's aspect-dependent lethal contour, plans one distance-only ingress on the grid, and scores it against one site. Its TODO comments point each task at the methods it needs. It runs as issued from a folder holding `FPToolkit.m` and the four CSV files. The appendix at the end of this handout has sample prompts for getting coding help from AI within the rules below.

Table 6: The `FPToolkit` methods. Positions are $[N\ E]$ in km; headings and bearings are degrees clockwise from North.

Method	What it returns	Task
<code>load</code>	the scenario struct <code>S</code> from the four CSV files	all
<code>signature</code>	the B-21 RCS or IR intensity at an aspect	T1
<code>chain</code>	$p_{\text{shot}}, \Delta s, \lambda$ (cued, autonomous), MEZ, LEZ per site	T2 check
<code>lethalRange</code>	a site's lethal radius toward an azimuth for a heading; with an SOJ orbit, the jammed radius	T1, T5
<code>orbitLegal</code>	whether an SOJ orbit is legal, and its clearance	T5
<code>grid</code>	the 5 km, 8-connected grid digraph	T3
<code>edgeRisk</code>	every edge's risk per site under a plan (SOJ victim and orbit, SEAD site, aspect-blind, pop-up)	T3, T5, T6
<code>route</code>	Dijkstra with this handout's edge cost and tie rule, for one w or a sweep; the route's per-site risk and P_S	T2, T3, T5, T6
<code>score</code>	length, per-site risk and P_S along any polyline (a straight line, a dogleg, a hop to the grid)	T6
<code>passiveDRMS</code>	theIRST network's CRLB DRMS at points or along a route	T4
<code>popup</code>	where the pop-up activates on a route, your heading ψ_0 there, and the leg and bearing points that follow	T6
<code>rwrCRLB</code>	your RWR's CRLB DRMS on an emitter from bearing points or along a leg	T6

Your Task

This project is individual. You may discuss the models and MATLAB with classmates; the analysis, the code you submit, and every word of the paper are yours.

T1 Threat picture (L32). Produce the laydown figure: every site's lethal contour for a **stated heading** (the one your ingress mostly flies is a good choice), MEZ and LEZ rings, and the EWR's coverage for the same heading. Nose-on and worst-case contours are optional: here the worst case is the MEZ everywhere, and nose-on is the MEZ for M1–M3. Name the gaps and the seams you see.

T2 Survival metric (L31). For each weapon site, go from the chain to $p_{\text{shot}}, \Delta s$ and λ (cued and autonomous) *by hand*, then check your numbers against `FPToolkit.chain`. Compute P_S along your T3 fuel-limited route, ingress and egress separately, and identify the binding threat on each leg.

T3 Route (L34). Build the grid digraph with aspect-aware edge costs, run Dijkstra, sweep w to the Pareto front, and make the fuel-limited choice (the choice is your code; the toolkit plans one leg at a time). Then plan the same mission aspect-blind: show how the route changes and what its true (aspect-aware) P_S is.

T4 Their track on you (L33). **One figure and one paragraph.** Compute the passive network's CRLB DRMS along your T3 fuel-limited (unsupported) route and plot it against distance flown. Does your route use a seam? Should it?

T5 Support package (L36). Evaluate every combination of SOJ victim (L1, M1, M2, M3), candidate orbit (O1–O5) and SEAD site (L1, M1, M2, M3), plus each asset alone, each re-planned exactly as in T3 under the same L_{max} . Report a compact table — SOJ victim by

SEAD site at each victim's best orbit, plus each asset alone — rather than all 105 plans. Pick one plan and defend it: its route, its ΔP_S over the unsupported route, the risk it leaves, and why it beats its nearest rivals. **Ties are possible:** each plan's route is unique, but two plans can give the same P_S . If yours ties, break the tie on a stated ground — for example the fuel margin, or which plan holds up better in your sensitivity. Show your victim's jammed contour for the heading your supported route flies through its strobe, with that route. Then **one sensitivity:** the victim's beamwidth *or* the SEAD factor. Does your pick survive? A flat result is a finding; explain it.

T6 Pop-up (L33/L35). Find where the pop-up activates on your T3 fuel-limited (unsupported) route and your heading ψ_0 there: the direction from the activation point to the first bearing point, 4 km on along the route (`FPToolkit.popup` returns it). Compare three ways to fly the next 60 km: **straight**, which follows your planned route, turns and all; the **left dogleg**, heading $\psi_0 - 30^\circ$ for 30 km, then $\psi_0 + 30^\circ$ for 30 km; and the **right dogleg**, heading $\psi_0 + 30^\circ$ for 30 km, then $\psi_0 - 30^\circ$ for 30 km. For each, report the CRLB DRMS of your RWR's bearing-only fix on the pop-up and the exposure it costs: the leg's risk with the pop-up in the threat set, and the track it gives up (the distance from the leg's end to your planned route's point 60 km on). Which would you fly, and why? *Optional:* re-plan the rest of the mission from the end of each leg with the pop-up in the threat set and compare route P_S ; an EKF (L35). Bookkeeping: take bearings every 4 km starting 4 km after activation (15 over the leg). Evaluate **J** at the prior position; the ± 10 km prior sets where you linearize and does not enter **J**. A re-plan starts from the grid node nearest the end of the leg; the hop to that node counts against fuel and risk.

R Recommendation and Limitations. The route to fly, with or without support, and the residual risk you accept. Then the limits of your models, including at least one L37 trend (for example GPS-denied navigation on the ingress, or a cognitive threat radar).

Optional sensitivity. Fly the signature package your Project 3 team recommended: subtract its band reductions (in dB) from the X and VHF columns of the pattern, apply its IR reductions to the IR column the way Project 3 did (the 40 W/sr skin floor and the hot parts above it each take their own reduction), and re-plan. Does the better signature change the route, or only its risk?

Use of AI. You *may* use AI tools to help write code from your own ideas and inputs — to turn an approach you have already worked out into MATLAB that calls the toolkit, to debug, or to tidy a figure. You may *not* hand the project to an AI: no pasting this handout in and asking for the solution, and no AI-written analysis, conclusions, or recommendation. You must **own and understand every line of code you submit** and be ready to explain any of it. The appendix shows what a good coding prompt looks like. The same rules extend to the paper. You may use AI to **typeset** your own work — to turn your text, equations, tables, and figures into LaTeX in the IEEE template, fix compile errors, and format BibTeX entries — and you may ask it to **point out** errors in your writing (grammar, unclear sentences, inconsistent notation) that you then fix yourself. You may *not* have AI write or rewrite your prose, analysis, conclusions, recommendation, or source vetting: the words and the judgments in the paper are yours. AI is not a source — every reference you cite must be one you opened and read. If AI was used in any way, a copy of (or a link to) every chat transcript is a required part of the code deliverable.

Course Citations and Sources

Your paper must cite **at least one course lesson from each block** — at least four lesson citations in all, for example L7, L14, L25 and L34 — and each must support a *specific* claim at the point where you make it (a model, a parameter’s meaning, a trade), not a list at the end. Cite a lesson in IEEE form, with the title written “Lesson NN: Title” (a colon) even though the course pages head each lesson with a dash:

[1] J. Jurado, “Lesson 34: Optimal Path Planning,” *ECE 495: Electromagnetic Warfare and Survivability*, U.S. Air Force Academy, Fall 2026. [Online]. Available: <https://usafa-ece.github.io/ece-495-ew/block04/L34.html>

External sources are optional. If you cite any (a textbook, a journal or conference paper, a report), list each one in Appendix A — Source Vetting of your paper: the source, what it supports, and one line of vetting (why it is trustworthy for that claim). **Publicly released sources only** — no classified, CUI or export-controlled material. Every scenario parameter in this project is notional.

The Paper

Write the project as a journal article in the IEEEtran journal format: two columns, **4–5 pages** excluding the references and appendices. Start from the issued template, `FinalProject_IEEE_Template.zip`. Overleaf is recommended (upload the zip as a new project); a local MacTeX install works too. `Final Project - Sample Paper.pdf`, on the course site, shows every section at the depth expected, on a notional toy problem: a model of format and depth, not an answer. Budget about 6 figures and 4 tables. Sections:

- Abstract and Index Terms
- I Introduction
- II Background and Related Work
- III Scenario and Threat Model
- IV Route-Planning Method
- V Results
- VI Discussion and Limitations
- VII Conclusion and Recommendation
- References
- Appendix A — Source Vetting (optional: only if you cite external sources)

Timeline

All submissions and draft exchanges go through the course Teams team.

- **L31:** project issued.
- **L35: checkpoint** (required; feedback only, not scored): a one-paragraph description of your approach, a first T1 laydown figure built from the starter’s, and the four lessons you plan to cite (one per block) with the claim each will support.
- **L38:** work day.

- **L39:** complete draft (PDF) due at the start of class; peer review #1 in class.
- **Before the start of L40:** peer review #2 (PDF).
- **L40:** revision work day.
- **L41, start of class:** final paper, response-to-reviewers table, code, and AI transcripts.

Deliverables

- **The paper** (PDF), in the IEEE template.
- **A response-to-reviewers table** (in the template): each reviewer comment, what you changed, and where — or why you did not.
- **Your code:** every script used for the analysis and the figures, runnable from the issued bundle (`FPToolkit.m`, the four CSV files and your scripts in one folder).
- **AI transcripts**, if any AI was used: a copy of or link to every chat, submitted with the code (see *Use of AI* above).
- **Two peer reviews** of classmates' drafts, on `Final Project - Peer Review Form.pdf`: review #1 in class at L39, review #2 before the start of L40.
- **Late deliverables** require coordination with the instructor at least 24 hours before the deadline; non-coordinated late submissions receive a zero.

Grading

Line	Points
Threat picture (T1, T4, T6)	15
Route method and cost design (T3)	20
Survivability, support package, sensitivity and recommendation (T2, T5, R)	20
Writing, figures and IEEE form	15
Course citations and sources (≥ 1 lesson per block)	10
Peer reviews given (specific, accurate, rubric-anchored)	10
Response to reviewers	5
Code reproducibility	5
Total	100

There is no innovation bonus: the Final Project is graded on the rubric alone.

Appendix: Getting Coding Help from AI — Sample Prompts

Each prompt below follows the *Use of AI* rules. You state your own approach first; you paste the toolkit's help text, so the AI calls the toolkit instead of inventing its own models; you ask for code that does exactly your approach; and you ask the AI to explain every line, so that you own it. None of them asks the AI to do the task, rank the options, or write your results. T2 is a hand calculation, so it has no prompt. Adapt the prompts to your own approach and your own variable names, paste the help output where the brackets say, and keep the transcript. The same prompts are on the Final Project page of the course site: copy them from there rather than from this PDF.

T1 — the laydown figure.

I am writing MATLAB for a course project. Two toolkit functions I will use are documented below.

[paste the output of: help FPToolkit.lethalRange]

[paste the output of: help FPToolkit.chain]

My approach: one map for a B-21 on a heading psi that I state. For each site in S.W, I draw its lethal contour $R = \text{FPToolkit.lethalRange}(S, \text{id}, \text{phi}, \text{psi})$ for $\text{phi} = 0:1:360$, plotted as $E + R.*\text{sind}(\text{phi})$, $N + R.*\text{cosd}(\text{phi})$, its MEZ as a dashed circle, and its LEZ (the LEZ_km column of $\text{FPToolkit.chain}(S)$) as a dash-dot circle. I also draw the EWR's coverage for the same heading with $\text{FPToolkit.lethalRange}(S, \text{"EWR"}, \text{phi}, \text{psi})$, and mark the IRST sites, the pop-up prior, and A, T and B. Please write a function $\text{plotLaydown}(S, \text{psi})$ that does exactly this with the toolkit functions, without re-implementing the radar model. Then explain what each line does, and tell me how to check one point of one contour by hand.

T3 — the fuel-limited choice.

I am writing MATLAB for a course project. The toolkit's route planner is documented below.

[paste the output of: help FPToolkit.route]

My approach: with $r = \text{FPToolkit.edgeRisk}(S, \text{Gr}, \text{struct}())$ and $w = [0 \text{ logspace}(0,5,31)]$, I sweep the ingress with $\text{rtIn} = \text{FPToolkit.route}(S, \text{Gr}, r, S.A, S.T, w)$ and the egress with $\text{rtEg} = \text{FPToolkit.route}(S, \text{Gr}, r, S.T, S.B, w)$. My rule for the fuel-limited choice: among all pairs (i, j) with $\text{rtIn}(i).L + \text{rtEg}(j).L \leq S.sc.Lmax_km$, take the pair with the smallest $\text{rtIn}(i).risk + \text{rtEg}(j).risk$; if two pairs tie in risk, take the shorter one. Write a script that makes the two sweeps with the toolkit, and a MATLAB function $[i, j] = \text{fuelChoice}(\text{rtIn}, \text{rtEg}, Lmax)$ that implements exactly my rule with plain loops. Explain each line, and give me a small test case with three routes per leg that I can check by hand.

T4 — their DRMS along my route.

I am writing MATLAB for a course project. The toolkit function below computes a passive network's CRLB DRMS.

[paste the output of: help FPToolkit.passiveDRMS]

My approach: myRoute.P holds my route's waypoints, and I call $[D, nIn, s] = \text{FPToolkit.passiveDRMS}(S, \text{myRoute.P}, 1)$ for 1 km steps. I want one figure: D against s on a log axis, with infinite values drawn at a 100 km cap, and the stretches where $nIn \geq 2$ and $D \geq 10$ km shaded. Write the code that makes this call and this plot, using only what passiveDRMS returns. Explain each line, including how the shading finds where each stretch starts and ends.

T5 — evaluating every support plan.

I am writing MATLAB for a course project. The toolkit functions I will use are documented below.

[paste the output of: help FPToolkit.edgeRisk]

[paste the output of: help FPToolkit.load]

My plan: loop over the victims ["L1" "M1" "M2" "M3"], over the rows of S.orbits, and over the SEAD options ["L1" "M1" "M2" "M3" "none"]. For each, build the plan struct, call FPToolkit.edgeRisk, and pass r to my own T3 function [PS, rtIn, rtEg] = planMission(S, Gr, r), which I have already written and tested. Store PS in a 4-by-5-by-5 array, and keep rtIn and rtEg for every plan in cell arrays of the same size so that I can plot any plan's route later. Then add the SEAD-only cases and the unsupported case. Write the loops, and a function that prints, for each victim, a table with the orbits as rows and the SEAD options as columns. Do not rank or pick plans: I will read the tables and decide. Explain each line.

T6 — the straight leg and the two given doglegs.

I am writing MATLAB for a course project. The toolkit functions I will use are documented below.

[paste the output of: help FPToolkit.popup]

[paste the output of: help FPToolkit.rwrCRLB]

I have pu = FPToolkit.popup(S, myIngress.P). The handout defines two doglegs from the activation point p0 = pu.p0, relative to my heading psi0 = pu.psi0: LEFT flies psi0 - 30 deg for 30 km, then psi0 + 30 deg for 30 km; RIGHT flies psi0 + 30 deg for 30 km, then psi0 - 30 deg for 30 km. Positions are [N E] in km and headings are degrees clockwise from North, so d km on heading h moves d*[cosd(h) sind(h)]. Write code that builds the waypoints [p0; w1; w2] for each dogleg, calls rwrCRLB with S.sc.rwr_spacing_km, and scores each leg with FPToolkit.score(S, leg, struct('popup', true)); for the straight leg, use pu.brg and pu.leg. For each of the three legs, print the DRMS, the leg risk, and the distance from the leg's end to pu.leg(end,:). Explain each line, and point out anything in my geometry that looks wrong.