GeneticWing

Learning to Soar:
Evolutionary Search in Learned Spaces, Physics-Informed Generative Planning and Distillation for Fixed-Wing UAV Soaring

Clément Dumeril

Research report, November 2026

Abstract

Fixed-wing drones can extend their endurance by harvesting energy from rising air, as sailplane pilots do, but planning energy-optimal trajectories through thermals is a non-convex, multi-modal problem whose good solutions sit against stall and load limits. This report studies how to make such planning both good and fast. A point-mass, differentially flat model of a 4.2 kg motor-glider (SB-XC), refined with a strip-theory wing that captures inner-wing stall in turns, turns any candidate trajectory into lift coefficient, bank and required thrust without integration. A slow offline *teacher* (a reactive orbit law tuned by a genetic algorithm, fitted and refined by sequential quadratic programming) solves 15,080 randomized single-thermal situations. A conditional variational autoencoder whose decoder outputs bounded *commands* flown by a differentiable simulator compresses those solutions into a 16-dimensional latent space; a genetic algorithm searching that space beats its own teacher by 0.84 energy units (cost -5.72 against -4.88 on 65 held-out situations), whereas the same algorithm on the raw 87-dimensional trajectory genome was the weakest optimizer tested. Flown in closed loop by an autopilot in turbulent air with model-predictive replanning, the advantage holds in calm air; the dominant loss is an error on the thermal's position. On 64 cross-country tasks of 20 km, a MacCready tactician whose climbs are flown by the latent genetic search climbs 61 % faster than a classic circling rule and matches the speed of motorized flight with 13 times less battery, without any stall once an inner-wing limiter is added to the autopilot. Finally, the search is distilled into a conditional flow-matching generator; after one DAgger round on the states met in flight, it plans in 0.24 s instead of 13 s and climbs at 2.23 m/s, as well as the search it imitates. The report details the models, algorithms and results, the failures that shaped them, and the path to a flight test on a laboratory airframe.

Contents

1 Introduction **4**
- 1.1 Motivation . . . 4
- 1.2 Research question and hypotheses . . . 4
- 1.3 Contributions . . . 4
- 1.4 Outline . . . 5

2 Background **6**
- 2.1 Soaring flight . . . 6
- 2.2 Differential flatness of the coordinated fixed wing . . . 7
- 2.3 Optimization . . . 7
- 2.4 Generative models . . . 7
- 2.5 Imitation and expert iteration . . . 8

3 Modelling **9**
- 3.1 Aircraft . . . 9
- 3.2 Point-mass dynamics . . . 9
- 3.3 The flat map . . . 10
- 3.4 Trajectory parameterization . . . 10
- 3.5 Energy cost . . . 10
- 3.6 Constraints and margins . . . 10
- 3.7 Strip-theory wing . . . 10
- 3.8 Integration: a lesson . . . 11

4 The teacher **12**
- 4.1 Phase 1: reproducing the ARPL glide . . . 12
- 4.2 Phase 2: which teacher? . . . 12
- 4.3 The teacher at scale . . . 13

5 Generative students and search in a learned space **14**
- 5.1 Flow matching on genomes . . . 14
- 5.2 The genetic algorithm in a latent space . . . 14
- 5.3 A decoder that is a simulator . . . 15
- 5.4 Realism: tight turns and the strip wing . . . 16
- 5.5 Sensitivity to the estimated polar . . . 16

6 Closed-loop flight **17**
- 6.1 Air, belief and autopilot . . . 17
- 6.2 Making replanning safe . . . 17
- 6.3 Results . . . 18

7 Cross-country flight **19**
7.1 Tasks . . . 19
7.2 A two-level pilot . . . 19
7.3 An inner-wing limiter . . . 20
7.4 Results . . . 20

8 Distilling the search **22**
8.1 Distillation . . . 22
8.2 A DAgger round . . . 22

9 Discussion **24**
9.1 What the results show . . . 24
9.2 Between MPC and reinforcement learning . . . 24
9.3 Threats to validity . . . 24

10 Towards a flight test **26**
10.1 Target platform . . . 26
10.2 Estimating the thermal in flight . . . 26
10.3 Integration . . . 26

11 Conclusion **27**

A Reproducibility **28**

B Main hyperparameters **29**

List of Figures

1.1 The GeneticWing pipeline. Green boxes are the learned components. 5

2.1 Thermal model: radial updraft profile at three altitudes, and Allen's vertical scaling. 6
2.2 SB-XC polar and MacCready speed-to-fly construction. 7

3.1 Forces in a banked, coordinated turn. The horizontal component of lift turns the aircraft; the inner wing flies slower than the outer one. 9
3.2 Strip theory in a turn. Each of 40 strips flies at its own airspeed; the slower inner wing reaches its section stall before the mean wing does. 11
3.3 Minimum steady turn radius against airspeed: point mass and strip-theory wing. 11

4.1 Phase 2: trajectories of the best methods on S1, S2 and S3, and convergence. . 13

5.1 Regression averages the two orbit directions into a straight line out of the thermal; flow matching samples circle in both directions. 14
5.2 Scaling: the student's gap to the teacher against the number of training situations. More data helps up to about 3,000 situations, then the limit is the model. 15
5.3 Latent search with the physics decoder: every candidate is decoded into commands and flown. 15
5.4 Energy gained on 77 test situations: the latent GA over the physics decoder beats the teacher. 16

6.1 Closed loop: plan in the belief, fly in the real air, replan every 10 s from the state reached. 17
6.2 Two 60 s flights in 3D, coloured by climb rate; orange: thermal core (15 and 30 m), blue dashes: plans, dotted: believed centre. Right: the spiral is clean but off-centre. 18

7.1 The cross-country pilot: a rule-based tactician over glide and climb modes, and the autopilot underneath. 19
7.2 A 20 km task in 3D: thermals as columns, climbs and glides of two pilots (vertical axis: altitude in m). 20
7.3 The same task: top view and barogram (altitude against distance toward the goal). 21

8.1 Distillation of the search and the DAgger loop. 22
8.2 Summary. Left: energy gained on 65 test situations. Right: mean climb rate on 64 tasks of 20 km. 23

List of Tables

1.1 Hypotheses and verdicts. 5

3.1 SB-XC model (C_{D0}, $C_{L,max}$ and Oswald factor are estimates, tested at ±20 % in Section 5.5). 9

4.1 Scenario S0: reproduction of the ARPL glide. 12
4.2 Phase 2: mean cost of feasible solutions (20,000 evaluations; lower is better). . 12
4.3 Re-seeding the teacher on 128 unsolved strip-wing situations (same margins). . 13

5.1 Phase 3: students on 78 held-out situations (teacher −5.19). 14
5.2 Phase 3, point-mass wing, 77 test situations (teacher −5.24). 15
5.3 Strip wing, retrained, 65 test situations. 16
5.4 Sensitivity to the SB-XC's estimated polar (65 situations, $d = 16$). 16

6.1 Closed loop, cluster run: 65 situations × 4 conditions, replanned every 10 s (no inner-wing limiter yet). 18

7.1 64 tasks of 20 km, protected autopilot for every pilot. 21

8.1 Distillation, 65 test situations (teacher −4.88). 22
8.2 64 tasks of 20 km, MacCready 0.5 / 1. 23

B.1 Main hyperparameters. 29

Chapter 1

Introduction

1.1 Motivation

Small fixed-wing UAVs are limited by their battery. Gliders show the alternative: the atmosphere is full of rising air (thermals, ridge lift, wind shear), and a pilot who knows where to circle can stay aloft for hours without a motor. Autonomous soaring has been demonstrated in flight, notably with the SB-XC sailplane at the US Naval Research Laboratory [5], and recent work on differentially flat fixed-wing planning [15, 16] has made energy-aware trajectory planning fast enough for onboard replanning, but stops short of harvesting energy.

Planning to soar is hard for three reasons. The objective (net energy over a horizon) is non-convex and multimodal: a thermal can be circled clockwise or counter-clockwise, entered from several sides, at several radii. The best trajectories sit on constraints (lift coefficient near stall, bank and load limits), so any approximation error makes them infeasible. And the problem must be solved again every few seconds in flight, because the air never matches the plan.

1.2 Research question and hypotheses

> *Can a generative model trained to imitate an offline optimizer propose near-optimal soaring trajectories fast enough for onboard replanning, once checked by the physics and refined by local optimization?*

The question was broken into six hypotheses, whose verdicts are summarized in Table 1.1 and argued in the following chapters.

1.3 Contributions

1. A flat, vectorized evaluation of soaring trajectories with a strip-theory wing, a unified energy cost and constraint handling for motor-gliders (Ch. 3).
2. Evidence that representation decides whether evolutionary search works: the same GA fails on raw trajectories and beats its teacher in the latent space of a physics-informed decoder (Ch. 5).
3. A command-space VAE whose decoder is a differentiable simulator, making every decoded plan dynamically feasible by construction (20 % to 100 % round-trip feasibility).
4. A closed-loop evaluation with an autopilot, turbulence and wrong beliefs, and the model-predictive ingredients that make replanning safe (shrinking horizon, warm start, state continuity) (Ch. 6).

Table 1.1: Hypotheses and verdicts.

Hypothesis	Verdict	Evidence
H1: a genetic algorithm (GA) is a good offline teacher	rejected, then reformulated	weakest optimizer on the 87-gene genome (Ch. 4); best method when searching a learned latent space (Ch. 5)
H2: a student plans in real time near the optimum	supported in the end	flow matching on genomes keeps ≈40 % of the gain; the distilled latent generator matches the GA in 0.24 s (Ch. 8)
H3: generative beats regression (multimodality)	supported	regression averages the two orbit directions into an infeasible straight line
H4: physics check + local refinement close the gap	partly	student-seeded NLP solves 76 % vs 8 % from straight flight
H5: robustness to wind errors and gusts	open	holds in calm air; a 15 m error on the thermal centre halves the gain (Ch. 6)
H6: self-improvement loop	supported once	first loop on position genomes: no gain; DAgger round on flight states: large gain (Ch. 8)

5. A two-level cross-country pilot (MacCready tactician, planned climbs) and an inner-wing stall limiter, evaluated on 64 tasks of 20 km (Ch. 7).

6. A distillation of the search into a flow-matching generator, corrected by DAgger on flight states, that plans in 0.24 s with the quality of the search (Ch. 8).

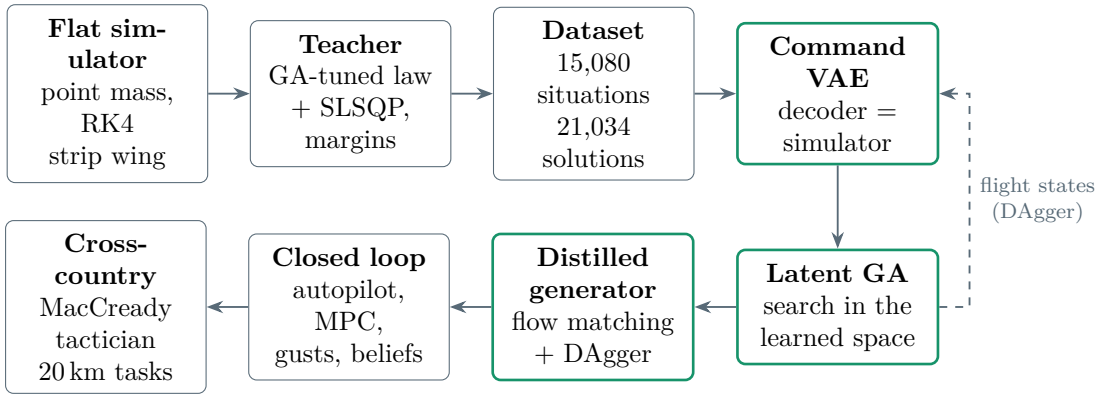


Figure 1.1: The GeneticWing pipeline. Green boxes are the learned components.

1.4 Outline

Chapter 2 reviews the background. Chapter 3 presents the physical model. Chapters 4 to 8 follow the six phases of the work: the teacher, the generative students and the latent search, the closed loop, cross-country flight, and the distillation of the search. Chapter 9 discusses the findings and their limits, Chapter 10 the path to a flight test, and Chapter 11 concludes.

Chapter 2

Background

2.1 Soaring flight

Thermals. A thermal is a column of warm rising air. Allen's model [1] scales the mean updraft $\bar{w}(z)$ and the outer radius $r_2(z)$ with the convective velocity $w^\star$ and the boundary-layer depth z_i:

$$\bar{w}(z) = w^\star\Big(\frac{z}{z_i}\Big)^{1/3}\Big(1 - 1.1\frac{z}{z_i}\Big), \qquad r_2(z) = \max\Big(10,\ 0.102\Big(\frac{z}{z_i}\Big)^{1/3}\Big(1 - 0.25\frac{z}{z_i}\Big)z_i\Big). \tag{2.1}$$

We use a Gaussian core of radius $r_2/2$ whose disk mean equals $\bar{w}$; the core drifts with the horizontal wind (Fig. 2.1). At 250 m altitude the core radius is about 30 m: a 15 m error on its position is large.

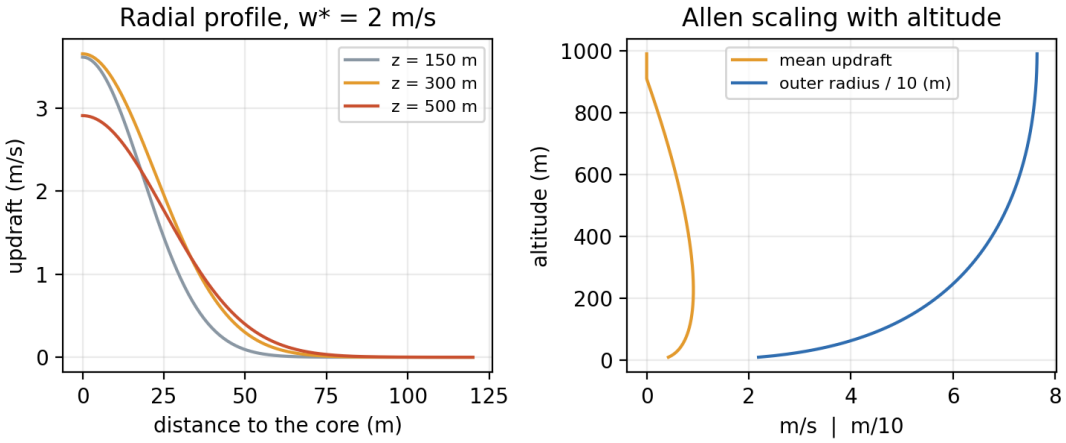


Figure 2.1: Thermal model: radial updraft profile at three altitudes, and Allen's vertical scaling.

MacCready theory. Cross-country pilots alternate climbs in thermals and glides between them. MacCready [13] showed that, expecting to climb at rate M in the next thermal, the airspeed that maximizes the average cross-country speed minimizes the time per unit distance including the climb:

$$V^\star(M, w) = \arg\min_V \frac{s(V) - w + M}{V}, \tag{2.2}$$

with $s(V)$ the still-air sink rate and w the vertical air motion. Geometrically, $V^\star$ is the tangent to the polar from the point $(0, M)$ (Fig. 2.2). The same reasoning tells the pilot to leave a thermal once its climb falls below M.

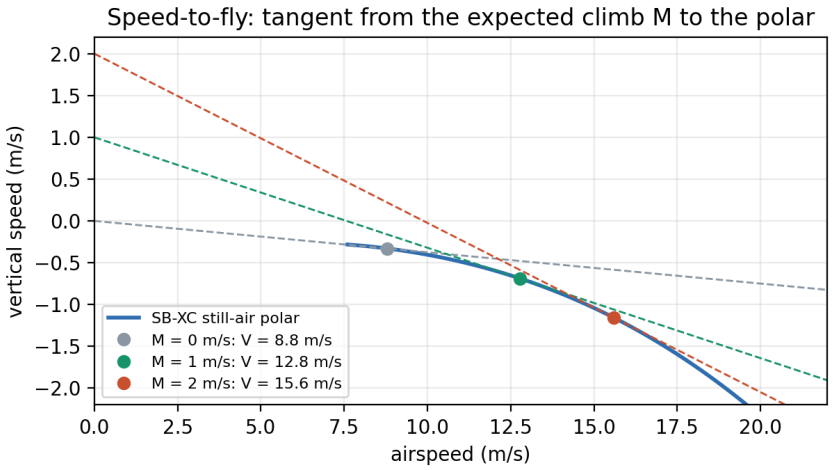


Figure 2.2: SB-XC polar and MacCready speed-to-fly construction.

2.2 Differential flatness of the coordinated fixed wing

For a point mass in coordinated flight with inputs (lift coefficient C_L, bank ϕ, thrust T), the position $\boldsymbol{p}(t)$ together with the wind $\boldsymbol{W}(\boldsymbol{p}, t)$ determines every other state and input algebraically [15]. This is the flat map used throughout (Section 3.3). Without a motor, position is no longer a flat output: a glider has two inputs for three outputs; the planner then imposes $T_{\text{req}} = 0$ as a constraint, which generalizes the sink-rate constraint of [16] to non-uniform wind.

2.3 Optimization

Genetic algorithms. A real-coded GA evolves a population of candidate vectors with simulated binary crossover (SBX) [3] and polynomial mutation. With constraints, Deb's feasibility rules [4] rank feasible solutions by cost and infeasible ones by total violation, feasible first. We also use the GA of T. Zohdi's course (80 % random immigrants per generation) to tune a 10-parameter reactive law, and CMA-ES [7] as a reference.

Local optimization. Sequential least-squares quadratic programming (SLSQP) [9] refines a candidate under constraints, with finite-difference gradients evaluated in one vectorized batch.

Constraint aggregation. Time-series constraints $g_k(t) \le 0$ are aggregated with the Kreisselmeier–Steinhauser (KS) function [10]

$$\mathrm{KS}(g) = \max_t g + \frac{1}{\rho} \log \sum_t e^{\rho(g(t) - \max_t g)}, \tag{2.3}$$

a smooth, conservative maximum (by at most $\log N/\rho$).

2.4 Generative models

Variational autoencoders. A conditional VAE [8] encodes a solution $\boldsymbol{x}$ given a context $\boldsymbol{c}$ into a latent $\boldsymbol{z}$ and decodes it back, trained on the evidence lower bound

$$\mathcal{L} = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})}\big[\log p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{c})\big] - \beta\, \mathrm{KL}\big(q(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{c}) \,\|\, \mathcal{N}(0, I)\big). \tag{2.4}$$

Flow matching. Conditional flow matching [11] with straight paths (rectified flow [12]) learns a velocity field v_θ that transports noise $\boldsymbol{x}_0 \sim \mathcal{N}(0, I)$ to data $\boldsymbol{x}_1$:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}_1} \left\| v_\theta\big((1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1,\, t,\, \boldsymbol{c}\big) - (\boldsymbol{x}_1 - \boldsymbol{x}_0) \right\|^2, \tag{2.5}$$

and samples by integrating $\dot{\boldsymbol{x}} = v_\theta(\boldsymbol{x}, t, \boldsymbol{c})$ from $t = 0$ to 1 in a few Euler steps. Because it models a distribution, it can represent both orbit directions instead of averaging them.

Latent variable evolution. Searching the latent space of a generative model with an evolutionary algorithm [19, 6] replaces a high-dimensional, coupled search by a low-dimensional one over plausible solutions.

2.5 Imitation and expert iteration

Behaviour cloning trains on the expert's state distribution and fails when the learner visits states the expert never did. DAgger [17] aggregates data on the learner's own state distribution, labelled by the expert. Expert iteration [2] and AlphaZero [18] alternate a slow search that improves on a policy and a network trained to reproduce the search's output. Chapter 8 applies both ideas to the latent genetic search.

Chapter 3

Modelling

3.1 Aircraft

Table 3.1: SB-XC model (C_{D0}, $C_{L,\max}$ and Oswald factor are estimates, tested at $\pm 20\,\%$ in Section 5.5).

Mass	4.22 kg	Wing area	1.07 m^2
Span	4.6 m	Aspect ratio	19.8
Max lift-to-drag ratio	27.1	Min-drag airspeed	8.05 m/s
$C_{L,\max}$	1.2	Motor (motor-glider conversion)	15 N
Propulsive efficiency	0.5	Polar	parabolic, $C_D = C_{D0} + kC_L^2$

3.2 Point-mass dynamics

In the inertial frame (east, north, up), with air velocity $\boldsymbol{v}_a = \boldsymbol{v} - \boldsymbol{W}(\boldsymbol{p}, t)$, airspeed $V_a = \|\boldsymbol{v}_a\|$ and dynamic pressure $q = \tfrac{1}{2}\rho V_a^2$,

$$m\dot{\boldsymbol{v}} = qSC_L\big(\cos\phi\,\hat{\boldsymbol{u}}_\perp + \sin\phi\,\hat{\boldsymbol{r}}\big) + \big(T - qS\,C_D(C_L)\big)\hat{\boldsymbol{v}}_a - mg\,\hat{\boldsymbol{e}}_z, \tag{3.1}$$

with $\hat{\boldsymbol{v}}_a$ the unit air velocity, $\hat{\boldsymbol{u}}_\perp$ the unit vector normal to it in the vertical plane and $\hat{\boldsymbol{r}} = \hat{\boldsymbol{v}}_a \times \hat{\boldsymbol{u}}_\perp$ the right-wing direction (Fig. 3.1).

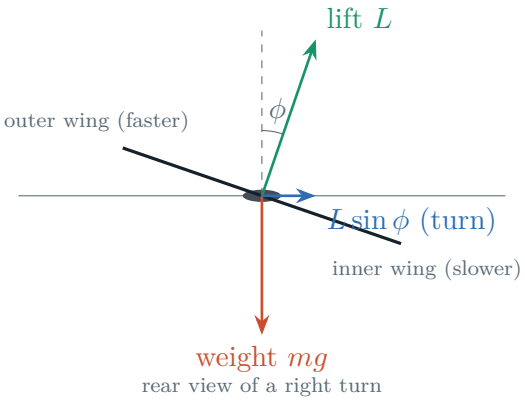


Figure 3.1: Forces in a banked, coordinated turn. The horizontal component of lift turns the aircraft; the inner wing flies slower than the outer one.

3.3 The flat map

Given a trajectory $\boldsymbol{p}(t)$ and its derivatives, $\boldsymbol{v} = \dot{\boldsymbol{p}}$ and $\boldsymbol{a} = \ddot{\boldsymbol{p}}$, the force the airframe must produce is $\boldsymbol{F} = m(\boldsymbol{a} + g\hat{\boldsymbol{e}}_z)$. Its component normal to $\boldsymbol{v}_a$ is the lift, and its axial component balances thrust and drag:

$$\boldsymbol{L} = \boldsymbol{F} - (\boldsymbol{F} \cdot \hat{\boldsymbol{v}}_a)\,\hat{\boldsymbol{v}}_a, \qquad C_L = \frac{\|\boldsymbol{L}\|}{qS}, \qquad n = \frac{\|\boldsymbol{L}\|}{mg}, \tag{3.2}$$

$$\phi = \operatorname{atan2}\left(\boldsymbol{L} \cdot \hat{\boldsymbol{r}},\ \boldsymbol{L} \cdot \hat{\boldsymbol{u}}_\perp\right), \qquad T_{\text{req}} = \boldsymbol{F} \cdot \hat{\boldsymbol{v}}_a + qS\,C_D(C_L). \tag{3.3}$$

No integration is needed: a population of trajectories, each sampled at 150 instants, is scored in one vectorized pass. $T_{\text{req}} < 0$ means excess energy that must be stored or bled.

3.4 Trajectory parameterization

Trajectories are clamped B-splines (chains of Bernstein polynomials) of degree 5 with 32 control points. The first three control points impose the initial position, velocity and acceleration; the 29 free ones are encoded "turtle-style" by a turn rate, a ground speed and a path angle per step, so that the genome has $3 \times 29 = 87$ genes with physical bounds (turn rate $\leq 30°$/s, climb angle $\leq 12°$, ground speed 4–22 m/s).

3.5 Energy cost

The cost is the net onboard energy over the horizon, normalized by the energy of level cruise on the motor at the best lift-to-drag speed:

$$J = \frac{1}{E_{\text{ref}}}\left(\int_0^H \frac{\max(T_{\text{req}}, 0)\,V_a}{\eta}\,dt \;-\; \frac{mg\,\Delta e}{\eta}\right), \qquad e = z + \frac{V_a^2}{2g}, \qquad E_{\text{ref}} = \frac{mg}{(L/D)_{\max}}\,\frac{V_{md}\,H}{\eta}. \tag{3.4}$$

$J = 1$ means cruising on the motor, $J < 0$ net energy gained from the air ($J = -5$: five times the energy of a minute of motor cruise gained in a minute).

3.6 Constraints and margins

Hinge functions, positive when violated, cover thrust and braking capacity, $C_L \leq 0.9\,C_{L,\max}$, airspeed, bank $\leq 45°$, load factor ≤ 2.5, altitude, mission area, and with the strip wing, section stall and aileron authority. The teacher solves with *tightened* constraints (constraint tightening, as in robust MPC): $C_L \leq 0.75\,C_{L,\max}$, bank $\leq 38°$, load ≤ 2.1, thrust $\leq 0.8\,T_{\max}$; its solutions are scored under the nominal ones. Optimal solutions sit on their active constraints, and without margins any imitation error breaks them.

3.7 Strip-theory wing

In a steady turn of radius R, a strip at spanwise position y flies at $V(y) = V(1 + y\cos\phi/R)$. Holding the bank requires the ailerons to add lift on the slower inner wing, so the inner sections reach their stall coefficient first, the ailerons can run out of authority, and their deflection adds drag. With 40 strips, the minimum steady turn radius at 9 m/s rises from 9 m (point mass) to 18 m, and the absolute minimum is about 11 m at 10.5 m/s (Fig. 3.3).

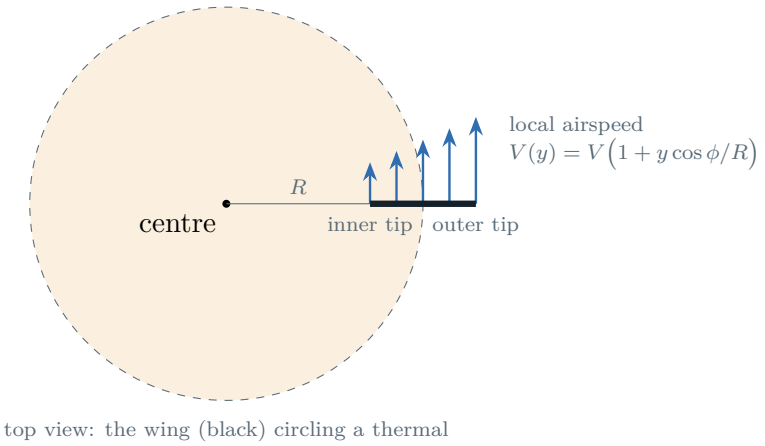


Figure 3.2: Strip theory in a turn. Each of 40 strips flies at its own airspeed; the slower inner wing reaches its section stall before the mean wing does.

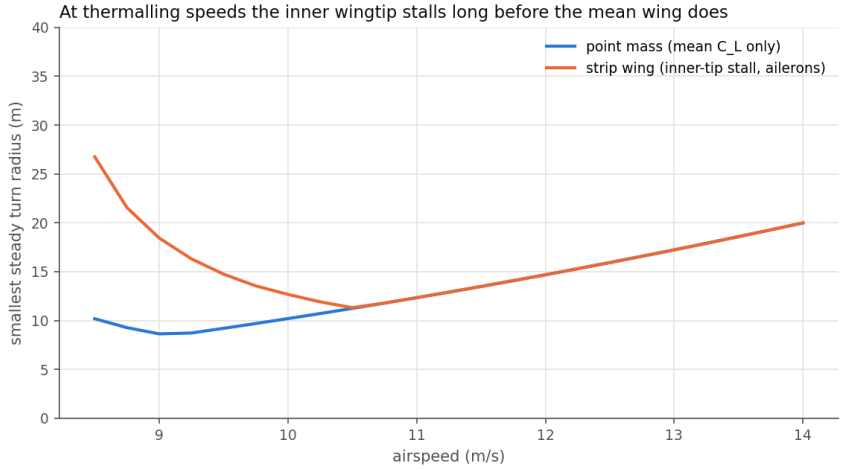


Figure 3.3: Minimum steady turn radius against airspeed: point mass and strip-theory wing.

3.8 Integration: a lesson

Simulated flights (the reactive law, the physics decoder, the closed loop) integrate equation (3.1). With explicit Euler, a step that turns the velocity by lift adds energy:

$$\|\boldsymbol{v} + \boldsymbol{a}\,\Delta t\|^2 = \|\boldsymbol{v}\|^2 + 2\,\boldsymbol{v}\cdot\boldsymbol{a}\,\Delta t + \|\boldsymbol{a}\|^2\Delta t^2, \tag{3.5}$$

and in a turn $\boldsymbol{v}\cdot\boldsymbol{a} \approx 0$, so every step injects $\|\boldsymbol{a}\|^2\Delta t^2$. A genetic search on the physics decoder found and exploited this free energy, reaching costs three times better than the teacher. All simulators now use the classical fourth-order Runge–Kutta scheme; a steady glide turn then loses exactly the polar's sink rate even with 0.4 s steps. *A powerful optimizer coupled to a simulator finds the simulator's flaws before it finds the physics.*

Chapter 4

The teacher

4.1 Phase 1: reproducing the ARPL glide

Scenario S0 reproduces the straight glide of [16]: a 400 m leg from 120 m with a 12.5 m/s airspeed reference and a 5.2 m/s tailwind, flown in closed loop through 50 Dryden turbulence realizations [14]. The error statistics of the real flight are recovered (Table 4.1); the planner converges from 20 of 20 perturbed initial guesses.

Table 4.1: Scenario S0: reproduction of the ARPL glide.

Metric	ARPL flight	Plan, uniform wind	Closed loop, Dryden
Mean airspeed (m/s)	12.50	12.50	12.61
Mean sink rate (m/s)	1.48	1.58	1.61
Glide ratio	9.83	11.15	11.27
Airspeed RMSE (m/s)	1.09	0.00	0.87
Netto RMSE (m/s)	0.69	0.00	0.71

4.2 Phase 2: which teacher?

Eleven methods were compared at equal budget on three scenarios (S1 single thermal, S2 thermal field to a goal, S3 ridge lift), three instances each (Table 4.2).

Table 4.2: Phase 2: mean cost of feasible solutions (20,000 evaluations; lower is better).

Method	S1 thermal	S2 field to goal	S3 ridge
SLSQP from straight flight	−2.77	infeasible	−2.11
Real-coded GA (tuned, population 50)	0.15	infeasible	−0.73
Zohdi GA	infeasible	infeasible	infeasible
CMA-ES + SLSQP	−3.80	infeasible	**−2.16**
GA, 100k evaluations	−0.99	0.47	−1.83
Reactive law (Zohdi-tuned)	−4.58	**−0.02**	−1.96
Law + SLSQP	**−6.04**	infeasible	−0.03

H1 is not supported as stated: on 87 coupled genes the GA is the weakest working optimizer. *Structure beats search*: a reactive orbit law (circle a centre at a given radius and speed, trade height for speed, motor below a floor) encodes the orbit the planners must otherwise discover. Fitting a genome to the law's flight and refining it by SLSQP gives the best teacher on S1 (−6.04, 30 % more energy than the law alone).

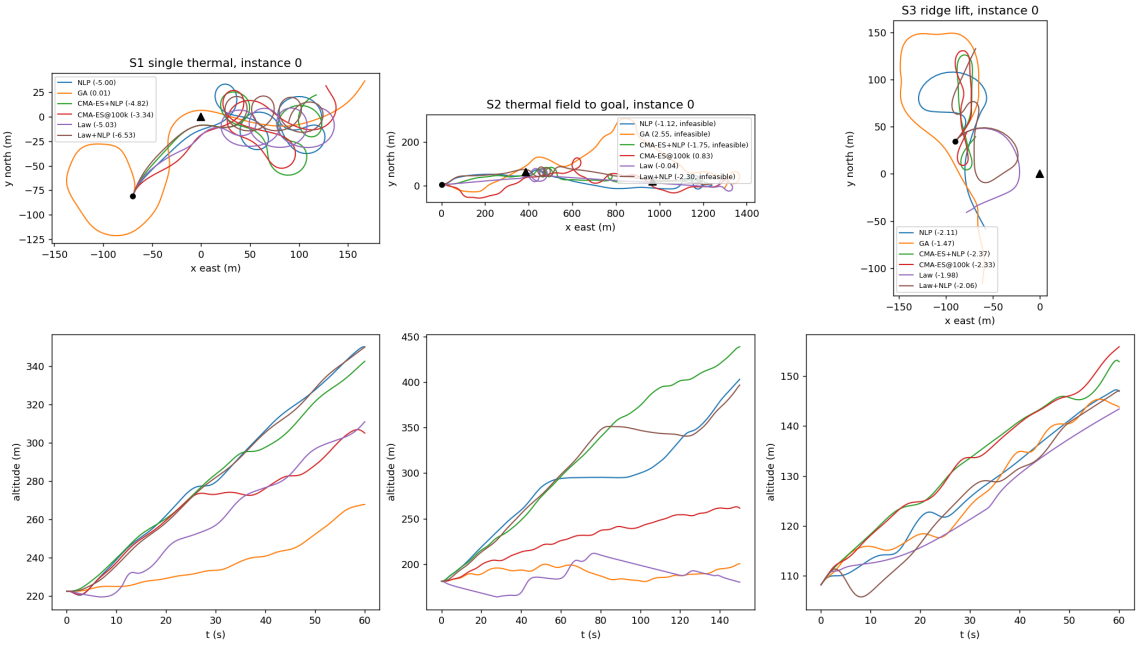


Figure 4.1: Phase 2: trajectories of the best methods on S1, S2 and S3, and convergence.

4.3 The teacher at scale

The randomized family S1 draws a thermal ($w^\star \in [1.2, 2.8]$ m/s), a wind (0–4 m/s, any direction) and a start (60–200 m from the thermal, 180–320 m altitude, any heading). Each situation is solved in both orbit directions. On a Slurm cluster, 21 nodes of 32 cores solve 15,080 situations in about 15 minutes.

Unsolved situations under the strip wing. With the strip wing and the margins, the teacher left 39 % of the situations unsolved. Since the family has a motor, feasible trajectories always exist: every failure is the optimizer's. In all 128 analysed cases the NLP ended with the inner wing stalled. Re-seeding the NLP with a *faster* orbit rescued 52 % of them at −4.63, while wider orbits and the straight glide were feasible but harvested nothing (Table 4.3). At a lower lift coefficient the slower inner wing keeps a stall margin. The two-seed teacher (tuned law and faster orbit, per direction) leaves 17 % unsolved; the faster orbit wins 83 % of the solutions.

Table 4.3: Re-seeding the teacher on 128 unsolved strip-wing situations (same margins).

Seed for the NLP	Feasible	Mean cost of those
Law with a faster orbit	52 %	−4.63
Straight glide	40 %	+0.91
Law with a wider orbit (80–120 m)	9–19 %	≈ +1
Same law, solved without margins	23–33 %	≈ −5.3

Chapter 5

Generative students and search in a learned space

5.1 Flow matching on genomes

A conditional flow-matching student (equation (2.5), MLP 4×512) learns the teacher's genomes from the situation's context, expressed in the aircraft's heading frame (invariance to rotation) and augmented by an exact left/right mirror. On 78 held-out situations (Table 5.1), regression never produces a feasible trajectory (H3), flow matching circles in both directions and solves 67–90 % of the situations in 30–50 ms, but keeps only about 40 % of the teacher's gain.

Table 5.1: Phase 3: students on 78 held-out situations (teacher −5.19).

Method	Solved	Cost	Gap	Time
Regression	0 %	–	–	1 ms
Flow matching, 64 samples, 8 steps	67 %	−2.06	3.11	49 ms
Flow matching + mutations + SLSQP	76 %	−2.51	2.72	1.0 s
Reactive law	100 %	−3.53	1.66	feedback
SLSQP from straight flight	8 %	1.49	6.95	0.9 s

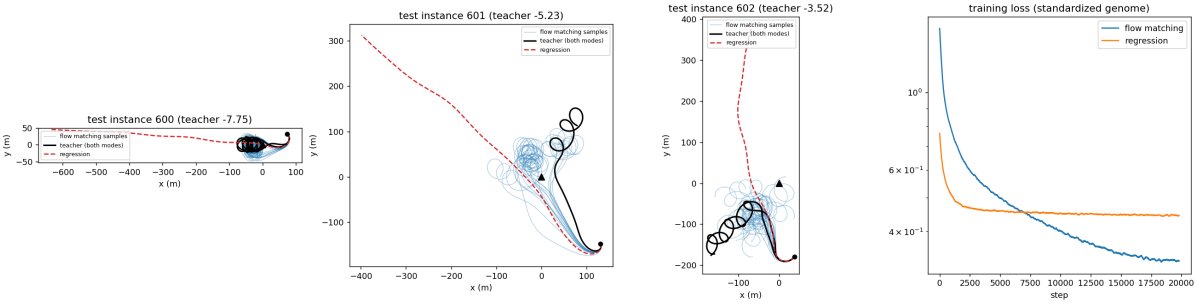


Figure 5.1: Regression averages the two orbit directions into a straight line out of the thermal; flow matching samples circle in both directions.

5.2 The genetic algorithm in a latent space

The phase 2 GA failed on 87 coupled genes. A conditional VAE compresses the teacher's trajectories into $d = 8$, 16 or 32 latent dimensions, and the same GA (population 50, 3,000 evaluations) searches that space; each individual is decoded and scored. Searching the latent

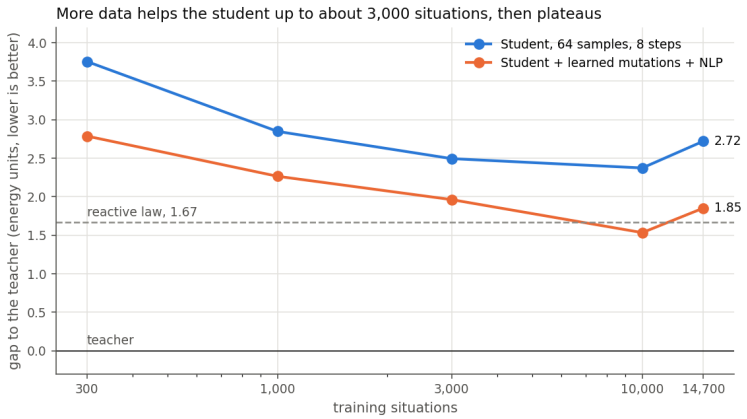


Figure 5.2: Scaling: the student's gap to the teacher against the number of training situations. More data helps up to about 3,000 situations, then the limit is the model.

space rescues the GA: 55 % to 100 % solved. But only about 20 % of the teacher's own trajectories survive the encode–decode round trip: small position errors become large C_L and thrust errors through the flat map.

5.3 A decoder that is a simulator

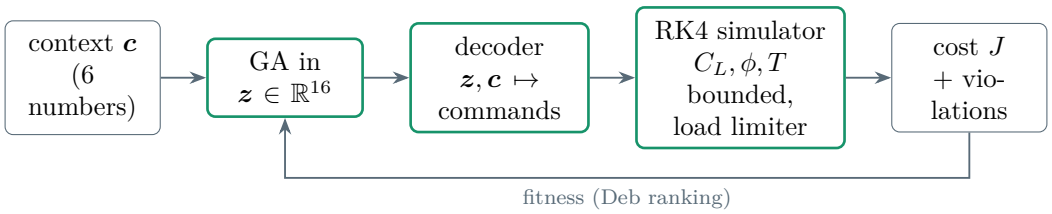


Figure 5.3: Latent search with the physics decoder: every candidate is decoded into commands and flown.

The decoder of the command VAE outputs *commands* (C_L, bank, thrust at 61 knots over 60 s) squashed into their limits, and a batched RK4 point-mass simulator (PyTorch, differentiable) turns them into the trajectory, with a load limiter that caps C_L at every step. C_L, bank, thrust and load factor cannot be violated by construction. The VAE learns the teacher's commands read off its trajectories by the flat map. The round trip goes from 20 % to 100 % feasible, and on 77 test situations (Table 5.2):

Table 5.2: Phase 3, point-mass wing, 77 test situations (teacher −5.24).

Method	Solved	Cost	Gap	Time per plan
GA on the 87 genes + SLSQP	55 %	+1.92	7.09	–
Latent GA, position VAE + SLSQP	100 %	−3.59	1.64	–
Teacher's commands, flown	100 %	−4.70	0.54	0.14 s
64 prior samples, physics VAE	100 %	−5.21	0.03	0.2 s
Latent GA, physics VAE	100 %	**−6.06**	**−0.82**	12.6 s

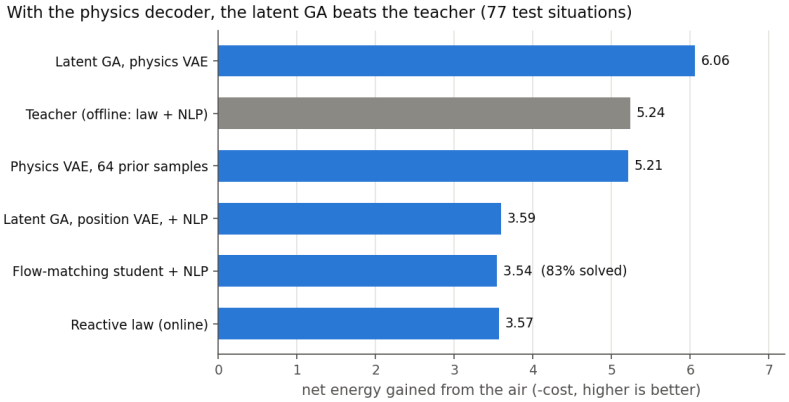


Figure 5.4: Energy gained on 77 test situations: the latent GA over the physics decoder beats the teacher.

5.4 Realism: tight turns and the strip wing

The latent GA gained mostly by circling tighter (median radius 15–25 m against 21–29 m). Under a common minimum turn radius its lead shrinks (from −0.85 without limit to −0.35 at 15 m) and reverses at 18 m; with the strip wing applied to every method and a VAE trained on point-mass trajectories, it falls behind the teacher (−4.28 against −4.70). Regenerating the teacher with the strip wing (two seeds, 15,080 situations) and retraining restores the lead (Table 5.3).

Table 5.3: Strip wing, retrained, 65 test situations.

Method	$d = 8$	$d = 16$	$d = 32$
Teacher	−4.88	−4.88	−4.88
Teacher's commands, flown	−4.45	−4.45	−4.45
64 prior samples	−4.72	−4.64	−4.65
Latent GA	−5.37	−5.69	**−5.72**

5.5 Sensitivity to the estimated polar

Scaling C_{D0}, $C_{L,\max}$ or the mass by 0.8 or 1.2 for every method (Table 5.4): drag moves everything by about 0.5 and leaves the gap at −0.8; the stall margin governs the lead. A heavier airframe or a lower $C_{L,\max}$ flies closer to stall and the lead shrinks to −0.1 (still positive); without search the prior does not transfer to another airframe.

Table 5.4: Sensitivity to the SB-XC's estimated polar (65 situations, $d = 16$).

Airframe	Teacher (solved)	Prior	Latent GA	Gap
nominal	−4.91 (100 %)	−4.64	−5.69	−0.79
$C_{D0} \times 0.8$	−5.48 (100 %)	−5.13	−6.29	−0.81
$C_{D0} \times 1.2$	−4.44 (100 %)	−4.24	−5.23	−0.79
$C_{L,\max} \times 1.2$	−5.30 (100 %)	−5.29	−6.24	−0.94
mass ×0.8	−5.70 (100 %)	−5.57	−6.93	−1.23
$C_{L,\max} \times 0.8$	−4.11 (89 %)	−2.53	−4.49	−0.09
mass ×1.2	−4.36 (100 %)	−2.81	−4.49	−0.13

Chapter 6

Closed-loop flight

6.1 Air, belief and autopilot

Until here every planner was scored in the air it planned in. In closed loop the two part (Fig. 6.1): the *air* is the situation with the strip wing plus Dryden turbulence; the planner has a *belief* of the thermal's centre, strength and wind, off by sampled errors. A path tracker (the autopilot) follows the plan as a path, not a clock: at each step it finds the closest reference point ahead and commands

$$\boldsymbol{a}_c = \boldsymbol{a}_r + k_v\, \boldsymbol{e}_v^{\perp} + k_p\, \boldsymbol{e}_p^{\perp} + \big(k_z e_z + k_V (V_a - V_{a,r})\big)\hat{\boldsymbol{e}}_z, \tag{6.1}$$

turned into C_L and bank by the flat map, through a first-order attitude lag (0.3 s), a bank-rate limit, a C_L protection at 90 % of $C_{L,\max}$, bank $\leq 40°$, the motor and the aileron drag.

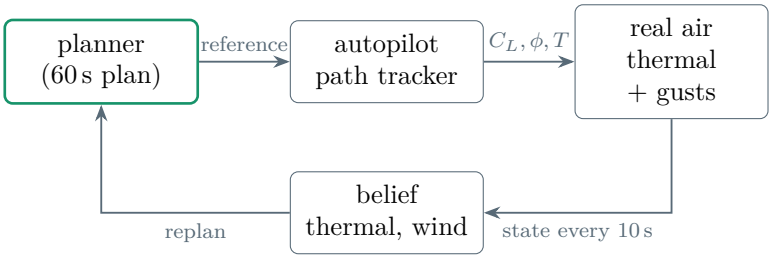


Figure 6.1: Closed loop: plan in the belief, fly in the real air, replan every 10 s from the state reached.

6.2 Making replanning safe

Replanning every 10 s with a planner that scores a fixed 60 s window failed: every plan put its energy gain at the end of its window, and the next replanning pushed that gain away (on one situation, −3.47 replanned against −7.11 open loop, with the first 10 s of every plan flown exactly as predicted). The fix is a *shrinking horizon*: a plan made at time t_0 of a flight of duration T_f minimizes

$$J_{t_0} = J\big(\boldsymbol{p}_{[t_0, T_f]}\big), \tag{6.2}$$

the cost up to the flight's end, so that replanning optimizes the objective that is scored. Every plan also starts from the state reached (velocity, acceleration, C_L, bank, thrust), keeps the orbit direction already taken, is warm-started from the rest of the plan being flown, and has that rest as a candidate: in its belief a new plan is never worse than carrying on.

6.3 Results

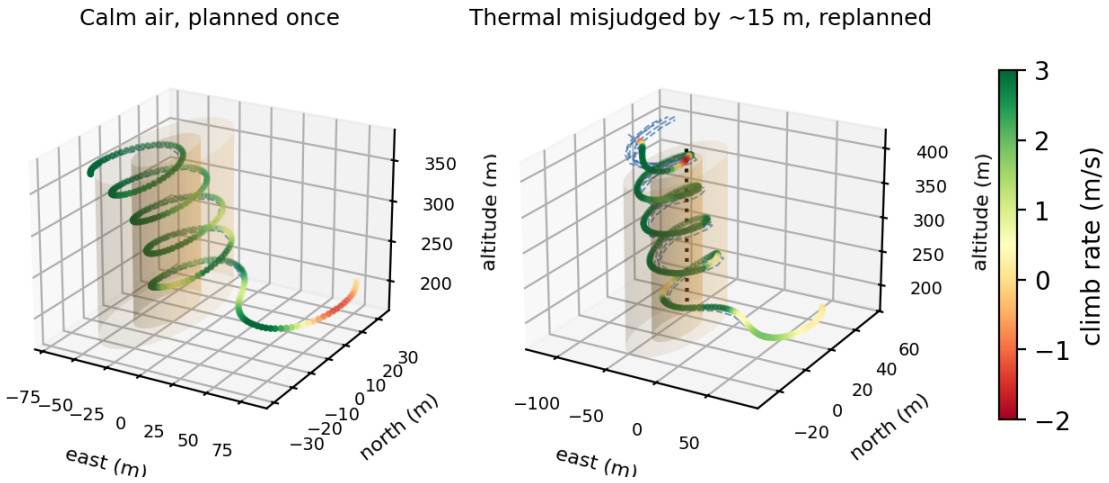


Figure 6.2: Two 60 s flights in 3D, coloured by climb rate; orange: thermal core (15 and 30 m), blue dashes: plans, dotted: believed centre. Right: the spiral is clean but off-centre.

Table 6.1: Closed loop, cluster run: 65 situations × 4 conditions, replanned every 10 s (no inner-wing limiter yet).

Condition	Planner	Flown cost	Without violation	Stalled
calm	teacher	−4.69	92 %	3 %
	latent GA	**−5.41**	55 %	11 %
	VAE prior	−4.56	86 %	3 %
turbulence	teacher	−4.15	25 %	52 %
	latent GA	−4.78	2 %	92 %
wrong belief	teacher	−2.03	37 %	35 %
	latent GA	−1.87	3 %	85 %
both	teacher	−1.87	3 %	85 %
	latent GA	−1.64	2 %	97 %

The autopilot tracks plans within 1–2 m on average (6.6 m at worst in gusts). In calm air the latent GA keeps its lead and replanning does at least as well as the open loop. A wrong belief (centre off by 15 m per axis, $w^\star$ by 20 %, wind by 0.5 m/s) costs more than everything else: every method falls to about −2, and replanning cannot help while the belief is not corrected (Fig. 6.2). Gusts make the inner wing stall: the plans use the whole stall margin.

Chapter 7

Cross-country flight

7.1 Tasks

A task sends the aircraft to a goal 20 km away through a Poisson field of drifting thermals (about one per km^2 over a 2 km corridor, $w^\star \in [1.2, 2.8]$ m/s), from 250–350 m, with a cloud base at 600 m and the motor allowed below 120 m. The score is equation (3.4) over the whole flight with E_{ref} the battery energy of the same distance on the motor (the motor scores about 1).

7.2 A two-level pilot

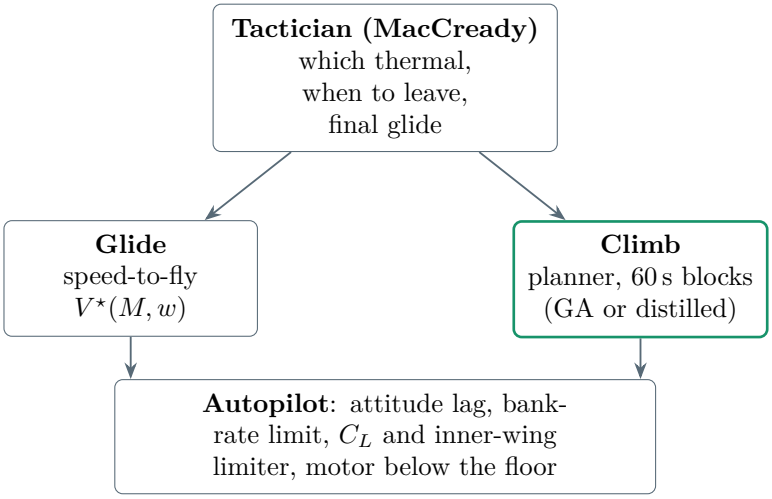


Figure 7.1: The cross-country pilot: a rule-based tactician over glide and climb modes, and the autopilot underneath.

The tactician chooses, among the thermals reachable with an arrival margin of 60 m above the floor, the one with the best rate of progress toward the goal, counting the time to climb back the height the glide costs:

$$i^\star = \arg\max_i \frac{\Delta_i}{d_i/V^\star + (d_i/(L/D)^\star)/c_i}, \tag{7.1}$$

with Δ_i the progress toward the goal, d_i the distance to thermal i, $(L/D)^\star$ the glide ratio at the speed-to-fly and c_i the expected climb rate. It leaves a thermal once the climb averaged over 20 s falls below M or at the cloud base, and starts the final glide when it is safe. Climbs

are flown either by a circling rule or by a planner in 60 s blocks (motor off), cut at the cloud base.

7.3 An inner-wing limiter

The first 64-task run showed the latent GA climbing 60 % faster but stalling a wing section in 19 to 69 % of its flights. A limiter in the autopilot caps the C_L command so that no strip passes a fraction $f = 0.9$ of its section stall, before and after the attitude lag:

$$C_L^{\mathrm{cmd}} \le \left. \frac{f\, c_{\ell,\max}}{\max_y c_\ell(y)/C_L} \right|_{V_a,\,\phi}, \tag{7.2}$$

the ratio of the local peak to the mean C_L being given by strip theory at the current airspeed and bank.

7.4 Results

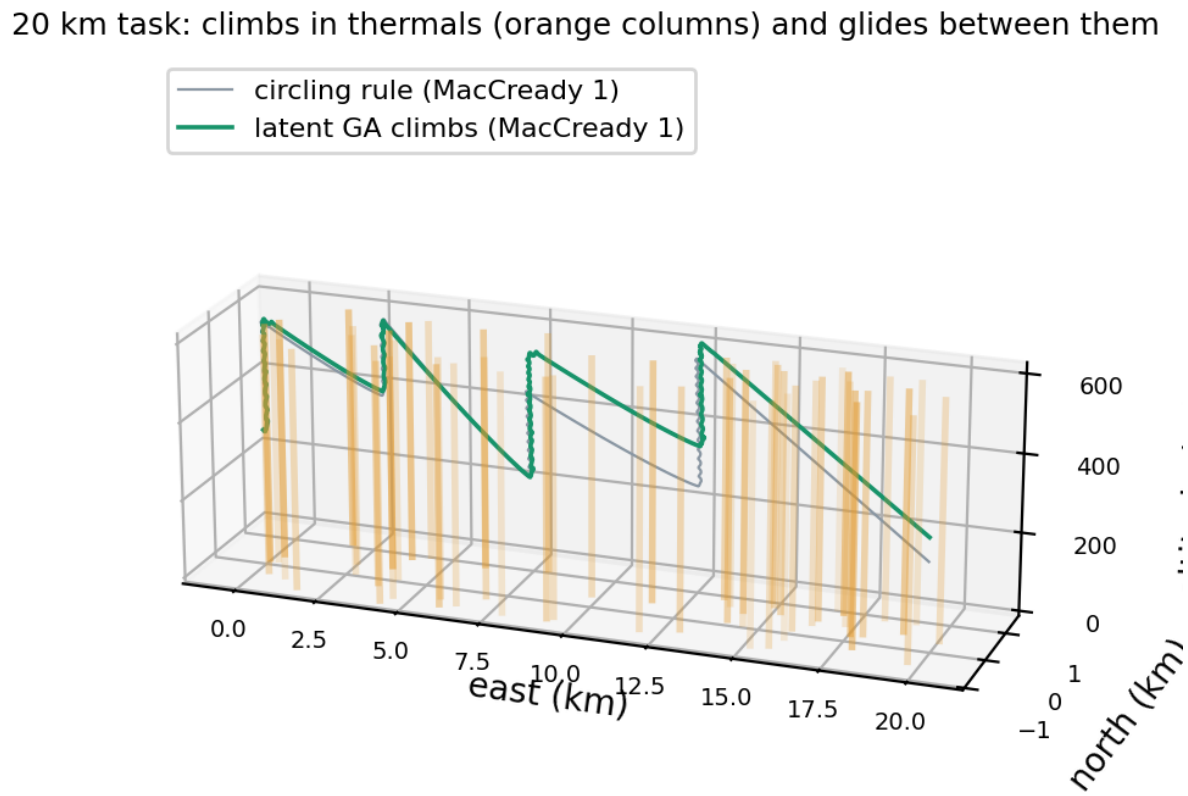


Figure 7.2: A 20 km task in 3D: thermals as columns, climbs and glides of two pilots (vertical axis: altitude in m).

The latent GA climbs 61 % faster than the circling rule; at the same setting the flight is 3.3 km/h faster on the same battery (faster on 89 % of the tasks), and at MacCready 1 it matches the motor's speed on 13 times less battery. The limiter removes every stall without

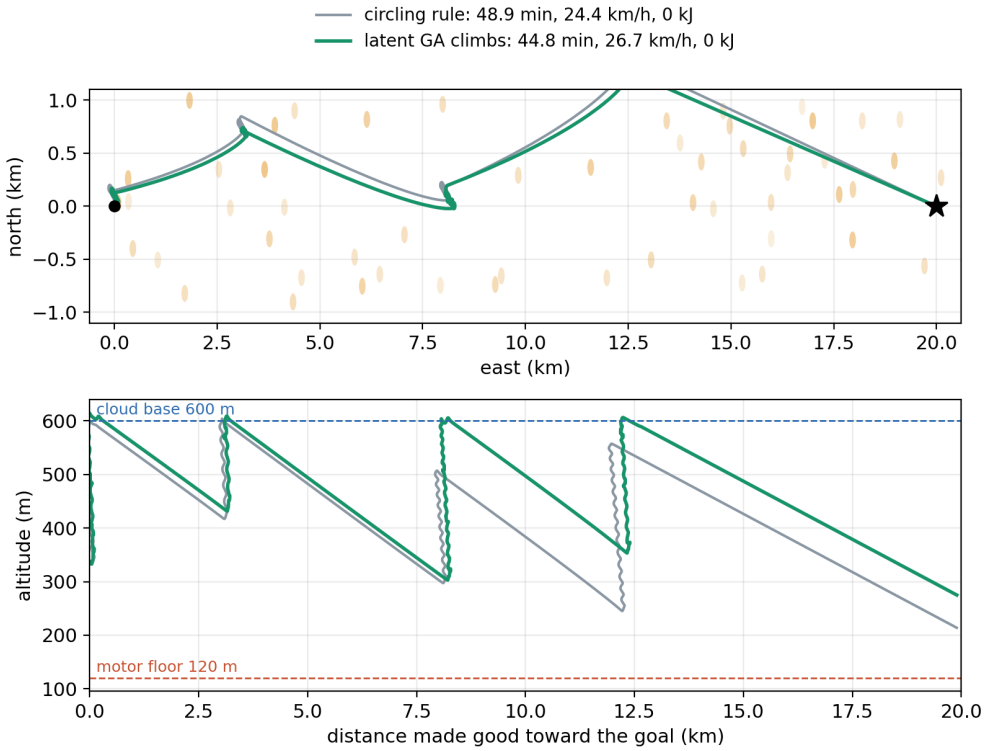


Figure 7.3: The same task: top view and barogram (altitude against distance toward the goal).

Table 7.1: 64 tasks of 20 km, protected autopilot for every pilot.

Pilot	Battery	Speed	Mean climb	Stalled	No violation
Motor all the way	61.9 kJ	29.2 km/h	–	0 %	100 %
Circling rule, MacCready 0.5	2.9 kJ	24.9 km/h	1.29 m/s	0 %	100 %
Circling rule, MacCready 1	19.3 kJ	28.3 km/h	1.22 m/s	0 %	100 %
Latent GA climbs, MacCready 0.5	2.9 kJ	**28.2 km/h**	**2.08 m/s**	0 %	98 %
Latent GA climbs, MacCready 1	4.7 kJ	**29.5 km/h**	**2.07 m/s**	0 %	100 %
GA with a 20 % stall margin, MacCready 1	4.7 kJ	29.0 km/h	1.90 m/s	0 %	100 %

reducing the lead (2.08 against 2.09 m/s): the gain did not come from flying at the stall. A stall margin in the plan is also safe but costs 6–8 % of the climb.

Chapter 8

Distilling the search

8.1 Distillation

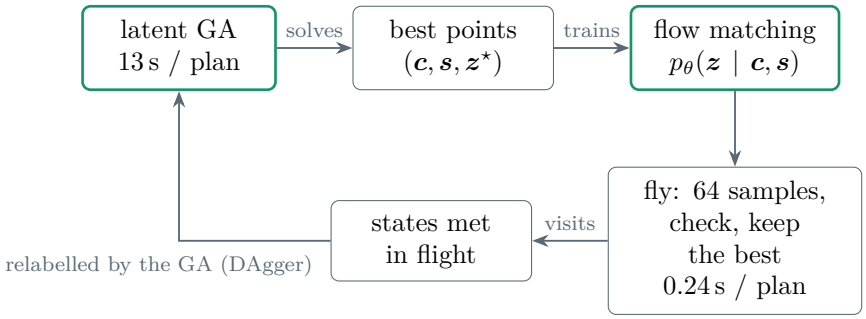


Figure 8.1: Distillation of the search and the DAgger loop.

The latent GA solves 3,000 training situations and keeps its best point $\boldsymbol{z}^\star$ in the command VAE's latent box; a conditional flow-matching model learns $p_\theta(\boldsymbol{z} \mid \boldsymbol{c})$ from those points (equation (2.5), MLP 3×256). In flight, 64 samples are decoded, flown in the belief and the best is kept (Table 8.1). The generator matches the GA at 57 times less compute and beats the teacher by 0.85; one sample is not enough, and a short search on top adds nothing.

Table 8.1: Distillation, 65 test situations (teacher −4.88).

Method	Solved	Cost	Gap	Time per plan
64 samples of the VAE prior	100 %	−4.64	+0.24	0.24 s
1 sample of the distilled generator	94 %	−3.52	+1.30	0.14 s
64 samples of the distilled generator	100 %	**−5.72**	**−0.85**	**0.24 s**
Distilled samples + a short GA (300 flights)	100 %	−5.59	−0.71	1.4 s
Latent GA (3,000 flights), the target	100 %	−5.66	−0.78	13.4 s

8.2 A DAgger round

Flown on the 20 km tasks, that generator kept little of the GA's lead (Table 8.2): it had learned level starts at 180–320 m, while a climb is replanned in a turn, up to 600 m, and its condition did not hold the current bank. Following DAgger, the GA flew the tasks with every climb plan it won recorded (1,095 plans: situation, aircraft state $\boldsymbol{s} = (\phi, V_a, C_L)$, latent point), and the generator was retrained on those plus the 3,000 situations of the first round, conditioned on $(\boldsymbol{c}, \boldsymbol{s})$.

Table 8.2: 64 tasks of 20 km, MacCready 0.5 / 1.

Climbs	Mean climb	Speed	Battery	Time per plan
Circling rule	1.29 / 1.22 m/s	24.9 / 28.3 km/h	2.9 / 19.3 kJ	feedback
Latent GA	2.08 / 2.07 m/s	28.2 / 29.5 km/h	2.9 / 4.7 kJ	≈13 s
Generator distilled on S1 only	1.50 / 1.17 m/s	25.1 / 28.8 km/h	4.2 / 20.2 kJ	0.24 s
Generator after one DAgger round	**2.23 / 2.20 m/s**	**28.5 / 30.2 km/h**	2.9 / 4.7 kJ	**0.24 s**

After one round the generator climbs as fast as the GA it imitates and slightly faster (faster on 84 % and 77 % of the tasks), likely because it draws 64 verified candidates from thousands of GA solutions while the GA has 1,000 flights per plan in flight. At MacCready 1 it beats the motor's speed (30.2 against 29.2 km/h) on 13 times less battery, with no stall, and it is unchanged on the S1 test situations.

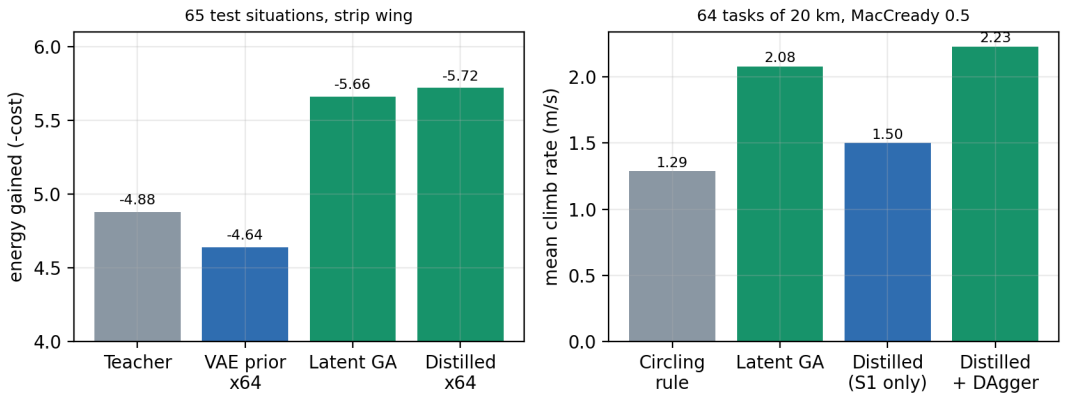


Figure 8.2: Summary. Left: energy gained on 65 test situations. Right: mean climb rate on 64 tasks of 20 km.

Chapter 9

Discussion

9.1 What the results show

1. **Representation decides whether evolutionary search works.** The same GA is the weakest optimizer on 87 coupled genes and the best method in a 16-dimensional learned space, beating the teacher that produced the data.
2. **Physics belongs in the decoder.** Decoding into bounded commands flown by a simulator turns feasibility from 20 % to 100 %.
3. **The binding limit of a circling sailplane is inner-wing stall**, not the turn radius; the cure is a faster orbit, and the stall margin (wing loading, $C_{L,\max}$), not drag, governs the gain.
4. **Receding horizons procrastinate** under a terminal-energy objective; plans must aim at the flight's end.
5. **Knowing where the thermal is matters more than planning well**: 15 m of error cost more than turbulence.
6. **Safety belongs in the autopilot**: a stall limiter costs nothing, a margin in the plan costs climb.
7. **Search can be distilled**, provided the generator is shown the states met in flight.

9.2 Between MPC and reinforcement learning

The final system is an amortized model-predictive controller: a learned proposal distribution replaces most of the online optimization, and a physics check keeps every plan honest. A GA over trajectories is planning (the MPC side); a GA over the 10 parameters of the reactive law is policy search (the reinforcement-learning side). The distilled generator sits between them: it is trained offline on search outputs like a policy, and used online as a sampler inside a planner.

9.3 Threats to validity

- Simulation only: a 3-DOF point mass with a strip-theory wing and an attitude lag, not 6-DOF.
- A synthetic thermal model; in the cross-country tasks the thermals are known exactly.

- One run per configuration for the cluster benchmarks (64–80 tasks each, paired comparisons).
- 17 % of the S1 situations are left unsolved by the teacher and excluded from the tests.
- The SB-XC's polar is estimated (tested at ±20 %).

Chapter 10

Towards a flight test

10.1 Target platform

The longer-term goal is to test the approach on one of the laboratory's airframes. The exact platform is to be confirmed with the laboratory (exchange with Jay); the sensitivity study tells what must be measured first: wing loading and $C_{L,\max}$, the parameters that govern the gain, before the drag polar.

10.2 Estimating the thermal in flight

The largest measured loss comes from the belief. The next step is an onboard estimator that recentres the thermal from the aircraft's own vario, in the spirit of ArduPilot's soaring controller: with $\hat{w}(t)$ the measured netto (air-mass vertical speed, from total-energy rate plus the polar's sink rate), a parametric thermal $w(\boldsymbol{p};\boldsymbol{\theta})$ with $\boldsymbol{\theta} = (x_c, y_c, w_{\text{peak}}, r)$ is fitted by an extended Kalman filter,

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + K_k(\hat{w}_k - w(\boldsymbol{p}_k;\hat{\boldsymbol{\theta}}_k)), \tag{10.1}$$

and the replanning uses the updated belief. Circling gives the filter the radial excitation it needs: the climb is stronger on the side closer to the core.

10.3 Integration

- **Software in the loop**: the planner (0.24 s per plan on one server core) inside PX4 or ArduPilot SITL, flying the existing autopilot loops.
- **Onboard compute**: the generator's 64 samples and their simulation are batched; an embedded GPU (e.g. Jetson class) should bring a plan well under the 10 s replanning period.
- **Retraining**: a teacher dataset and a command VAE for the identified airframe, then the DAgger round on its own flight states.

Chapter 11

Conclusion

Planning to soar is a hard, multimodal, constraint-bound problem. Searching the right space turned a failing genetic algorithm into a planner that beats the optimizer that taught it; putting the physics in the decoder made every proposal feasible; flying the plans in closed loop showed that knowing where the thermal is now matters more than planning well; and distilling the search, then correcting it on the states met in flight, gave a planner that runs in a quarter of a second with the quality of the search. On 64 tasks of 20 km it climbs 70 % faster than a classic circling rule, matches motorized flight on 13 times less battery, and does not stall. The next steps are an onboard thermal estimator and a flight test on a laboratory airframe.

Appendix A

Reproducibility

All code is public at https://github.com/clementdumeril/GeneticWing. The main commands (on the cluster, from the repository root) are:

- `bash slurm/submit_all.sh strip`: teacher on 15,080 situations, merge, physics study;
- `bash slurm/submit_all.sh sensitivity`, `closedloop`, `xc`, `distill`, `dagger`: the studies of Chapters 5–8;
- `python scenarios/phase4_view.py`, `phase5_view.py`: self-contained 3D pages of flights.

Compute: a Slurm cluster with CPU nodes of 32 cores and A40 GPUs; two desktop machines (i7-3770, i5-6500) for smaller runs. The software uses NumPy, SciPy and PyTorch, with 69 unit tests.

Appendix B

Main hyperparameters

Table B.1: Main hyperparameters.

Genome	B-spline degree 5, 32 control points, 87 genes
Teacher	law seeds (tuned, faster orbit) $\times$ 2 directions, SLSQP 6,000 evaluations, margins
GA	real-coded, population 50, SBX + polynomial mutation, Deb ranking
Latent GA	3,000 flights (1,000 in the cross-country climbs)
Command VAE	61 knots $\times$ 3 commands, latent $d = 16$, $\beta = 0.1$, MLP 3×512, 20,000 steps
Flow matching	MLP 3×256 (latent) or 4×512 (genomes), 20,000 steps, 8 Euler steps
Simulator	RK4, 0.4 s steps (plans), 0.02 s (tracker)
Closed loop	60 s flights, replanning every 10 s, shrinking horizon
Cross-country	20 km, cloud base 600 m, motor floor 120 m, climbs in 60 s blocks
Limiter	every wing section below 90 % of its stall

Bibliography

[1] M. J. Allen, "Updraft model for development of autonomous soaring uninhabited air vehicles," AIAA Aerospace Sciences Meeting, AIAA 2006-1510, 2006.

[2] T. Anthony, Z. Tian, D. Barber, "Thinking fast and slow with deep learning and tree search," NeurIPS, 2017.

[3] K. Deb, R. B. Agrawal, "Simulated binary crossover for continuous search space," Complex Systems, 9(2), 1995.

[4] K. Deb, "An efficient constraint handling method for genetic algorithms," Computer Methods in Applied Mechanics and Engineering, 186, 2000.

[5] D. J. Edwards, "Implementation details and flight test results of an autonomous soaring controller," AIAA Guidance, Navigation and Control Conference, 2008.

[6] A. Gaier, A. Asteroth, J.-B. Mouret, "Discovering representations for black-box optimization," GECCO, 2020.

[7] N. Hansen, A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," Evolutionary Computation, 9(2), 2001.

[8] D. P. Kingma, M. Welling, "Auto-encoding variational Bayes," ICLR, 2014.

[9] D. Kraft, "A software package for sequential quadratic programming," DFVLR-FB 88-28, 1988.

[10] G. Kreisselmeier, R. Steinhauser, "Systematic control design by optimizing a vector performance index," IFAC Symposium on Computer-Aided Design of Control Systems, 1979.

[11] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, M. Le, "Flow matching for generative modeling," ICLR, 2023.

[12] X. Liu, C. Gong, Q. Liu, "Flow straight and fast: learning to generate and transfer data with rectified flow," ICLR, 2023.

[13] P. B. MacCready, "Optimum airspeed selector," Soaring, 1958.

[14] U.S. Department of Defense, MIL-F-8785C, "Flying qualities of piloted airplanes," 1980.

[15] L. Morando et al., "Trajectory planning and control for differentially flat fixed-wing aerial systems," ICRA, 2025 (arXiv:2502.00581).

[16] L. Morando, S. Bobbili, C. Masci, G. Loianno, "Wind-aware optimal trajectory planning for efficient gliding of fixed-wing aerial systems," arXiv:2605.15619, 2026.

[17] S. Ross, G. J. Gordon, J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," AISTATS, 2011.

[18] D. Silver et al., "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," Science, 362, 2018.

[19] V. Volz, J. Schrum, J. Liu, S. M. Lucas, A. Smith, S. Risi, "Evolving Mario levels in the latent space of a deep convolutional generative adversarial network," GECCO, 2018.