Passing Every Gate Without Learning: A Deterministic Dead-Reckoning Policy with Monocular Gate Corrections for AI Grand Prix Virtual Qualifier 1

AI-GP Autonomy Team

Abstract

We describe the policy that cleared every gate of Virtual Qualifier 1 (VQ1) of the Anduril AI Grand Prix. The policy contains no learned component. It is a fixed function of the admissible observation stream: body-frame velocity and attitude are integrated into a position estimate, a surveyed six-gate course artifact supplies gate geometry, and a geometric position and velocity controller emits collective-thrust-and-body-rate (CTBR) commands at 50 Hz. Because open-loop integration of velocity drifts without bound, the estimate is corrected by three independent channels driven by the monocular gate detection: an absolute position fix from apparent gate size, a proportional body-rate correction from gate bearing, and a gate-index resync driven by the elapsed-time-since-last-gate signal. We report the architecture, the correction channels, the deploy boundary that keeps privileged simulator state out of the loop, and the specific reason this design does not survive contact with Virtual Qualifier 2, whose contract withdraws both attitude and gate telemetry.

1 Problem

VQ1 asks for an autonomous flight through a fixed, surveyed course in the FlightSim simulator. Completion requires passing every gate in order without contacting a gate or the environment. The course is released in advance, so gate geometry is a legitimate prior rather than something to be discovered online.

Two design routes were available. The first is to learn a policy, by reinforcement learning against the simulator or by behaviour cloning from a privileged expert. The second is to write the controller directly, treating the released course as a map and the simulator's admissible telemetry as a partial state estimate. We took the second route for VQ1, for three reasons.

Diagnosability. Every failure in a hand-written stack has a location. When the vehicle clipped a gate, the diagnostic trace named the correction channel and the gate index responsible, and a parameter existed to change it.

Attempt budget. Qualification attempts are scarce. A deterministic policy produces the same trajectory from the same start state, so an improvement measured offline is an improvement in qualification, and a passing run is reproducible rather than resampled.

Time. The course does not change between attempts. Generalization has no value in VQ1, and specialization to one course is cheaper to get right than a policy that has to hold across courses it will never see.

Table 1: The legal observation frame. Fields are grouped by source; the camera image is carried alongside the 24-element vector.

Field	Meaning	Dim
imu		
`ang_vel_body`	body angular rates	3
`lin_acc_body`	body linear acceleration	3
ego		
`speed_body`	body-frame velocity	3
`attitude`	roll, pitch, yaw	3
gate		
`visible`	detection confidence	1
`center_uv`	gate centre in image	2
`size_uv`	apparent gate extent	2
`bearing`	bearing to gate	2
history		
`prev_action`	previous CTBR command	4
`time_since_last_gate`	gate timer	1

2 The admissible interface

The policy consumes a flat 24-dimensional observation frame, concatenated with the camera image. Table 1 gives the layout. Nothing else is read. In particular the policy never touches `info["state"]` or any scorer or oracle channel exposed by the simulator for evaluation.

Two entries in this table carry most of the design. `ego.attitude` is a free, drift-free orientation, which removes the hardest part of inertial navigation. `gate.*` is a detection of the active gate, which anchors position against a surveyed landmark. Both are withdrawn in VQ2 (Section 6).

The action is $u = [\, f,\ \omega_x,\ \omega_y,\ \omega_z \,] \in [-1, 1]^4$, collective thrust and body rates, published every 20 ms.

Table 2: Controller and vehicle constants used for VQ1.

Quantity	Value
Position gain	1.5
Velocity gain	2.9
Attitude gain	6.0
Target speed	$10\,\mathrm{m\,s^{-1}}$
Max desired acceleration	$15\,\mathrm{m\,s^{-2}}$
Post-gate lookahead	3.0 m
Hover thrust action	−0.414
Max body rate, xy	$10\,\mathrm{rad\,s^{-1}}$
Max body rate, z	$6\,\mathrm{rad\,s^{-1}}$
Control period	0.02 s
Action smoothing α	0.7

3 Method

3.1 Position by dead reckoning

Attitude and body-frame velocity are given, so the only quantity that must be integrated is position. At each step the body velocity is rotated into the world frame by the reported attitude and integrated with a fixed step $\Delta t = 0.02\,\mathrm{s}$:

$$v_w = R(\phi, \theta, \psi)\, v_b, \qquad p_{k+1} = p_k + v_w\, \Delta t. \tag{1}$$

This is a first-order integrator on a signal the simulator already reports, so the error is dominated by accumulated velocity bias rather than by attitude error. It still diverges: over a full course the unaided estimate walks far enough to miss a gate. The three correction channels below exist entirely to bound that walk.

3.2 The course prior

The released course artifact stores six gates as NED positions with orientation quaternions, plus a racing line. It is content-addressed by hash, so a policy manifest names exactly one course revision and a run cannot silently execute against a different survey.

Gate geometry is used for two things: the target that the controller flies toward, and the anchor that converts a gate detection into an absolute position.

3.3 Geometric control

Given the position estimate, the active gate centre, and a lookahead past the gate plane, a geometric controller computes a desired acceleration from position and velocity error, saturates it, converts it to a desired thrust direction, and reduces the attitude error to body rates. The tuning used for qualification is in Table 2.

3.4 Correction channel 1: absolute position from apparent size

The gate outer dimension is known from the course artifact. The detector reports the gate's apparent extent in normalized image units, so range follows directly from the pinhole relation

$$\hat{r} = \frac{W_{\mathrm{gate}}}{\max(s_u, s_v)}. \tag{2}$$

Back-projecting the detected centre at that range gives a camera-frame vector to the gate, which is rotated through the fixed camera-to-body extrinsic and the reported attitude into a world-frame offset. Subtracting it from the surveyed gate centre yields an absolute position measurement:

$$p_{\mathrm{vis}} = c_{\mathrm{gate}} - R(\phi, \theta, \psi)\, R_{cb}\, d_c. \tag{3}$$

The estimate is not replaced by this measurement. It is moved a fraction of the way toward it, with the vertical component scaled separately and the whole step clamped to a maximum magnitude per tick. Gating on detection confidence, on a valid range window, and on a minimum age of the current gate keeps a single bad detection from teleporting the estimate.

This is the channel that makes dead reckoning viable. It converts a monocular detection of a known-size landmark into a position fix, and it fires on every frame in which the active gate is visible and plausible.

3.5 Correction channel 2: visual servoing on bearing

The position fix corrects the estimate. It does not correct a controller that is aiming at a stale estimate during the final approach, when the gate fills the frame and small lateral error is what decides contact. A second channel therefore adds a proportional body-rate term computed directly from the image, bypassing the estimator:

$$e_{\mathrm{lat}} = 0.75\, b_x + 0.25\, u_x, \tag{4}$$

a blend weighted toward the reported bearing over the raw image centre. The term is added to the controller's roll and pitch rate commands with per-gate gains, is capped in magnitude, and activates only above a confidence threshold and after the current gate has been active for a minimum time. Its effect is to centre the gate in view during the commit phase regardless of what the position estimate believes.

3.6 Correction channel 3: gate-index resync

The most damaging failure is not positional. It is bookkeeping: if the policy believes it is flying gate 4 while the simulator scores gate 3, every subsequent target is wrong and the run is lost. Three mechanisms keep the index honest.

Estimated crossing. The policy checks its own estimated position against the active gate plane with a margin and advances when it believes it has crossed.

Timer resync. `time_since_last_gate` resets when a gate is actually scored. That reset is an admissible, unambiguous event signal, and it is authoritative over the estimate. On observing one, the policy snaps its gate index to the timer's, subject to a confirmation window, an advance delay, and a clamp that prevents the estimate from running ahead of the observed timer.

Visual loss. A gate that was tracked and then disappears from the frame at close range is evidence of a pass, used as a fallback when neither of the above has fired.

The three are ordered, and the timer wins, because it is the only one grounded in the scorer's own accounting rather than in the policy's belief.

3.7 Safety and smoothing

Two further stages sit between the corrected command and the vehicle. A close-gate safety stage detects a large, centred gate at short range and reduces thrust and yaw authority, trading time for clearance where a contact would end the run. An exponential filter with $\alpha = 0.7$ then smooths the action, which suppresses the rate chatter that the bearing correction can introduce when detection confidence flickers.

Nearly every threshold above accepts a per-gate override. This is the practical core of exact-course specialization: gate 1 is approached from a standing start, gate 3 requires a late vertical bias, gate 4 needs a yaw-rate limit, and the manifest records each as an explicit named parameter rather than as an implicit property of learned weights.

4 Determinism and the deploy boundary

The policy is deterministic in a strong sense: no network, no sampling, no stochastic exploration, no wall-clock dependence. From a given start state and a given manifest it produces the same command sequence every time. This is what made a scarce attempt budget workable, because an offline improvement is an improvement in qualification rather than a change in a distribution.

Legality is enforced structurally rather than by convention. The policy reads a fixed-width slice of the observation vector and nothing else; the sibling progression policy rejects privileged constructor arguments by name, refusing `gate_positions`, `start_pose`, `track`, and `course_path` outright. Privileged data is confined to a quarantined diagnostic path used for offline analysis, which is never consulted at inference. A geometric controller that does read simulator state exists in the codebase, but as a teacher for offline work; the deployed policy instantiates the same control law against its own estimate.

5 Result

The policy passed Virtual Qualifier 1, clearing all six gates in order. Each promoted candidate is archived as an immutable manifest naming the course hash, the controller, and every tuned parameter, so a passing configuration can be replayed exactly.

6 Why this does not transfer to VQ2

VQ2 blocks four inbound streams: `ATTITUDE`, `LOCAL_POSITION_NED`, `ODOMETRY`, and `GATE_INFO`. Every load-bearing element of the design above depends on at least one of them.

Dead reckoning assumed a free, drift-free attitude and a reported body velocity; without them, position must come from visual-inertial estimation with all the drift that implies. The absolute position fix assumed a gate detection supplied by the simulator; the detector must now be ours, trained on the actual image distribution. Even the gate timer, the one signal we trusted above our own estimate, is no longer available in the same form.

The correct reading is not that the VQ1 policy needs porting. It is that the paradigm it belongs to, plan against a survey and track it with a state estimate, is inadmissible under the VQ2 contract. Our successor work therefore abandons the map entirely and guides on bearings alone, using optical looming for closing rate; it is described separately.

7 Limitations

Three are worth stating plainly.

The policy generalizes to nothing. It encodes one course at one simulator revision, and a resurvey invalidates the manifest.

The tuning surface is large, and the per-gate overrides that made the run reliable were found by search over attempts rather than derived. A different operator would not arrive at the same numbers.

The design is only sound because gate detections were provided. We did not solve gate perception in VQ1; we consumed it. That debt is what VQ2 collects.