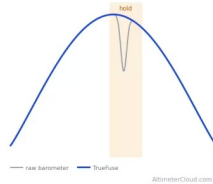


TrueFuse

Sensor fusion for altimeters

The barometer is the reference. The inertial sensors carry it across the moments it cannot be believed.



The reference for the altitude fusion filter that runs on AltimeterCloud and on our altimeters: what it does, phase by phase, and how to read what it did on your flight. For the story of how it came about, the three articles cover it: [TrueFuse 2C](#) for the design and why we did not build a Kalman filter, [TrueFuse 2E](#) for the slew witness and the port to the Nano, and [TrueFuse 2F](#) for the port coefficient and the upward bridge.

VERSION	RUNS	INPUTS	APOGEE	FAILURE MODE
TrueFuse 2F, TruePath 6	Live on Mercury and Jupiter, at save on the Nano	Barometer, accelerometer, gyroscope	Always the barometer; the raw figure is kept in every log	Decline to the raw barometer

Competition, record flights & apogee

Our filter doesn't invent an apogee. At apogee the rockets speed is low and the most accurate sensor is the barometer. TrueFuse uses the accelerometer and gyroscope information to repair clear pressure errors, not adjust the maximum altitude.

Altitude competition rules are written around the barometer, and TrueFuse was designed with that in mind. The FAI Sporting Code (Section 4, Volume S, 4.9.2.1 and Volume EDIC, Section 2) requires an electronic altimeter to use a barometric measurement technique, to record the flight altitude as the difference between the peak pressure altitude and the pad, to convert pressure to height on the standard atmosphere, to sample at not less than ten samples per second, and to be fully enclosed within the airframe, with an accuracy of 2% or 2 metres, whichever is greater. The NAR's US Model Rocket Sporting Code (20.2) says the same: barometric measurement of apogee above the launch pad, converted on the ICAO or US Standard Atmosphere, and only data from an altimeter on the Contest Board's approved list. Neither allows an altitude that came from an accelerometer.

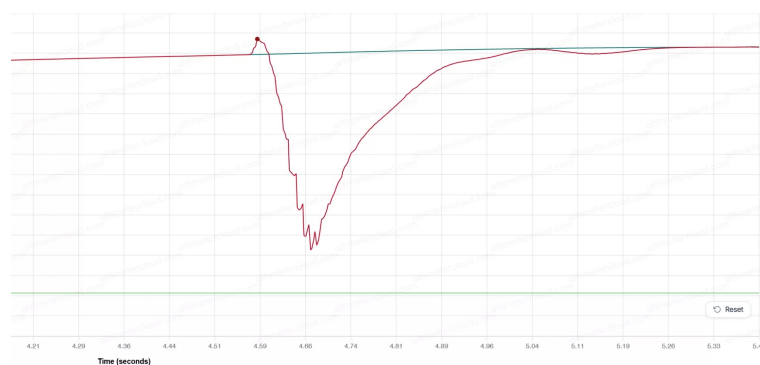
This is how the fusion sits against those rules:

- **Apogee is a barometric measurement.** The inertial channel only carries the altitude while the rocket is fast. Its weight falls to zero as the rocket slows toward apogee, and at apogee the handover is hard: the accelerometer retires and the peak is read from the barometer. The inertial sensors never set an altitude; they only carry one between two barometric anchors.
- **The raw barometric altitude is always kept.** Every log carries the unfiltered pressure altitude in its own column, exactly as recorded, converted from pressure with the standard-atmosphere formula and never written to afterwards. The raw apogee and the processed apogee are both in the header and side by side on the flight page.
- **The filter cannot raise an altitude the barometer did not support.** Holds and bridges carry the output on the barometer's own fitted trend across an artifact, and every guard resolves to the raw barometer. There is no path by which the processed apogee exceeds what the pressure sensor measured outside an artifact.
- **What the filter changes is marked.** Each hold, bridge and fuse window is recorded with its start and end, so an official can see exactly where the processed trace departs from the raw one and why.

The many ways a barometer lies

A barometric altimeter measures static pressure and converts it to height. For most of a flight it is the best altitude sensor on the board: it does not drift, it needs no starting condition, and it is immune to the errors that pile up in an integrated accelerometer. For a few seconds of the same flight it is badly wrong, and the errors are not noise. They are structured, physical and tied to specific events.

- **The launch window.** While the rocket is fast, air moving past the static port changes the pressure it sees. The error is a sustained bias with the sign and size of the airflow, so averaging only gives a smoother version of the same wrong number. It can start before the rocket moves, when the motor coming up to pressure washes over the pad, and on some airframes the port unwinds far enough to read tens of metres below the pad while the rocket is climbing.
- **Ejection.** A charge pressurises the airframe and the port reads a step that has nothing to do with altitude: a dive of tens or hundreds of metres in a fraction of a second, then a recovery. Main deployments, delay grains venting and landing impacts do smaller versions of the same thing. On an airframe where the altimeter bay is sealed from the charge, the same event can read the other way up, as a small rise.
- **Everything else.** Transonic effects around the airframe, poorly sized or blocked vents that turn a smooth flight into a staircase, and sunlight flashing onto the sensor die through the port as the rocket swings under its parachute.



An ejection charge firing just before apogee. The red line is what the barometer recorded. The green line is the path TrueFuse carried the rocket along, on the trend it was following before the charge went off.

An accelerometer knows nothing about air pressure and everything about thrust, but integrating it twice turns a small bias into a large altitude error within seconds. Neither sensor is good enough alone. TrueFuse uses the barometer as the reference and the inertial sensors only to carry the altitude across the moments where the barometer has been shown to be untrustworthy. It is not a smoother. It is a referee.

Why it runs live

An altimeter that trusts pressure alone has one safe response to a pressure event: stop believing itself. The conventional answer is a lockout, a blanking period in which altitude-driven decisions are suspended until the sensor settles. Lockouts work, every sensible flight computer carries them, ours included. But a lockout does not fix the altitude, it just agrees not to look at it for a while, and the blanking lands at the busiest moment of the flight with a length chosen in advance for the worst charge the design might ever see.

TrueFuse removes the reason for the blanking. The altitude is carried across the event rather than abandoned, so the vehicle still has a believable height and vertical speed while the port is recovering, and the window ends when the barometer genuinely agrees again rather than when a timer runs out. Air brakes, staging and altitude-triggered outputs keep running through the disturbance, and the lockouts become what they should always have been: a fallback layer rather than the plan.

It is strictly causal. Every decision uses samples that had already been taken, and every revision is checked for that by re-running the flight battery with the log cut off at four points: the output up to each cut has to be identical to the full run, to the last digit.

Marked, not hidden. Every hold and bridge is recorded with its start and end, and the log header says what the filter did and whether it declined. Anything that should still behave cautiously during a disturbance can be told exactly when one was open. Caution becomes a choice rather than the only option.

Where it runs

The deployment matches each device's job. The Mercury and the Jupiter fly the filter live, one sample at a time inside the flight loop, because they act during the flight. The Nano has no outputs and nothing to decide in the air, so it runs the same filter over the recorded flight when the log is saved, which costs nothing that matters and lets it do one thing the live devices cannot: revise its own past. The website runs the same law again on every uploaded flight.

Whichever device it is, the porting rule is absolute: a device runs the same law, not a resemblance of it. Each port is a statement-for-statement transcription of the reference, proven against it on the same logs sample by sample. The Nano's saved column sits on the reference within two millimetres at its worst sample. The Mercury's live fusion differs from the reference by 1.4 mm at its worst sample on the flight used to prove it, with the apogee identical.

The whole machine

Nothing in the chain is tuned to a flight, and nothing estimates a quantity that can be calculated. Conditioning happens before any fusion decision, the inertial vertical channel produces its own altitude and velocity, and a supervisor that knows the flight's phases decides, sample by sample, which witness to believe.

TrueFuse 2F



AltimeterCloud.com

TrueFuse 2F in flight order. The guards along the right can reseed or veto any stage, and every one of them resolves to the same safe action.

The flight, phase by phase

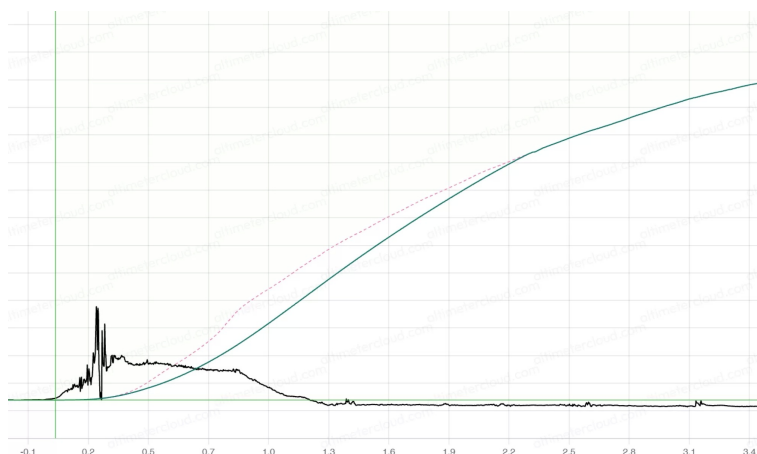
On the pad

The filter waits for genuine stillness and learns three things at once: the direction of gravity in the sensor's own frame, the gyroscope's at-rest bias, and the accelerometer's scale. The output is held still while the rocket sits on the rail. The pad stage ends the moment the airframe has been out of the still band for longer than any hold may last, whatever the launch stamp says, because a device can stamp launch late and a pad hold that is still live twenty-five metres up pins the output to zero.

A slip watch runs through the pad wait. A loosely mounted altimeter that moves on the pad invalidates the vertical it has just learnt, so when the slip watch sees it move, the calibration is reseeded and a line on your flight page says so.

The launch fuse

From launch, the attitude carry rotates the learnt vertical through every rotation the airframe makes, the thrust vector is projected onto it, and the projection is integrated into an inertial velocity and altitude. A climb guard stops the output falling while the accelerometer says the rocket is climbing. Thrust is a second attitude witness in its own right: early in a burn it utterly dominates everything else the accelerometer sees, and comparing its direction with the pad's gravity line gives the rail angle, measured entirely inside the sensor's frame. A loose altimeter cannot fake either one, which is what makes their disagreement such a reliable sign of a unit that has shifted at ignition.

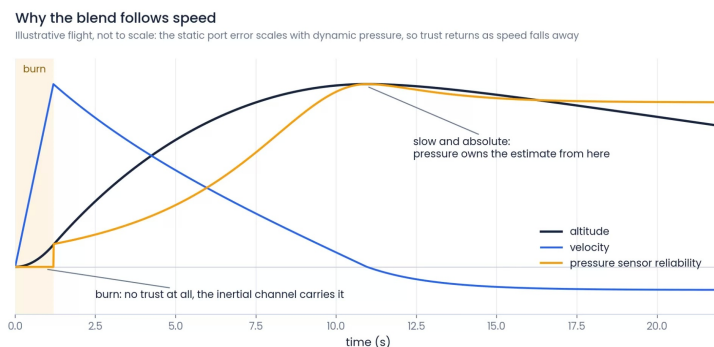


Ignition on a loose altimeter: the accelerometer noise as it slips to a new seat, the red dashed raw altitude disturbed by the thrust, and the green TrueFuse output on what the physics says was possible.

A fuse that fires and then decides it was a false start forgets everything it learnt. On the website and on the Nano, which have the samples in an array, the output printed between the false launch and the change of mind is rewritten back to pressure; the header counts those as false-trigger windows unwound. A live device cannot revise what it has already written, so that mechanism does not exist on the Mercury or Jupiter, by design rather than omission.

The blend, scheduled on speed

The port's error is aerodynamic: it scales with the dynamic pressure of the air moving over the port, so it is largest at top speed and melts away as the rocket slows. Speed is therefore not a convenient proxy for trust, it is very nearly the size of the barometer's lie. And the two witnesses age in opposite directions: pressure is the absolute, drift-free reference, wrong only while the air is moving fast; the inertial altitude is superb through the burn and accumulates error with every second that passes.



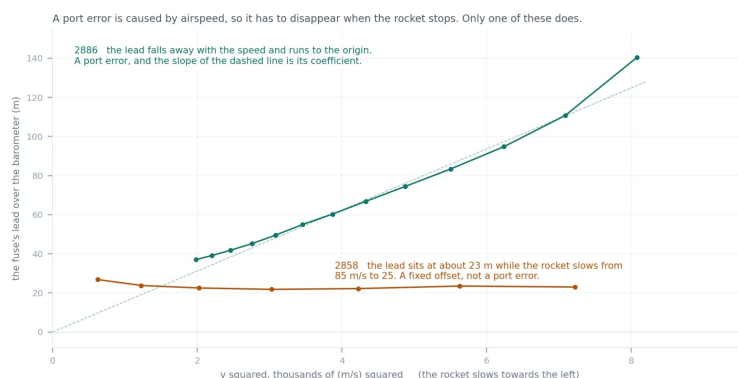
The blend in one picture: the port's reliability collapses exactly when the rocket is fast, and returns in full as it slows toward the apogee it is about to measure.

So the schedule follows the physics. At high speed the inertial channel carries the estimate. As the rocket decelerates toward apogee the weight returns to the barometer, which by then has become both honest and absolute, at precisely the moment the

flight's headline number is about to be written. The weight slews in both directions on an eased curve, front-loaded and tangent at the join, so the output never arrives on the barometer at an angle. The merge band is anchored at the first sample of the blend, and the speed used for the schedule is the filter's own isolated estimate, never the fused output, because a schedule that reads the thing it controls is a feedback loop wearing a schedule's clothing.

The port coefficient

The blend schedules around the port error. Since 2F the filter also measures it. A port error scales with the square of airspeed, so it has a coefficient with units: metres of altitude error per (metre per second) squared. Starting a hundred milliseconds after burnout and running for four hundred more, the filter takes its own inertial altitude, subtracts the raw pressure altitude and divides by the square of its own velocity, as a trimmed mean with the highest and lowest tenth thrown away. It needs at least ten samples in the window, a velocity above thirty metres per second, and the trend sensor not owning the output. The number is a measurement of that airframe, that port and that flight, and it is reported on the flight page as the port coefficient k.



A real port error is a line heading for the origin when the fuse's lead over the barometer is plotted against the square of the speed, and its slope is the coefficient. A lead that does not fall away with the speed is something else wearing the same clothes.

The correction is added to the pressure the blend hands back to, and only while five things are true at once: the rocket is climbing, the top has not been reached, no bridge is open, the raw is below the fuse's own line, and the trend sensor has been quietly in track for a full second. On most flights it never fires. A coefficient at or below zero means the barometer was not lagging in the way the mechanism exists to fix, and the flight is then identical to one without it. A mechanism that knows when to leave well alone is worth as much as one that knows when to act.

Apogee, and the descent

At apogee the handover is hard: the inertial channel retires and the schedules switch, while the trend guard stays armed. This is deliberate. On descent a rocket swings under its parachute, and the accelerometer reads the whole pendulum: line snatch, canopy breathing, oscillation and spin, a soup of genuine forces with almost no honest vertical story in it. Fusing that in would not improve the altitude, it would manufacture one. Past apogee, pressure is the only witness worth believing, and the descent belongs to the barometer, guarded against artifacts but never embellished.

Holds and bridges

Two mechanisms guard the pressure stream from the coast to the landing, because delay charges and main deployments do not politely wait for apogee. Both ride the flight's own recent physics rather than handing over to the accelerometer, because many pressure events assault both witnesses at once: an ejection charge that spikes the port also fires mechanical shock through the accelerometer, and the snatch loads that follow a deployment are violent, structured noise on exactly the sensor you might hope to lean on.

The trend sensor: holds

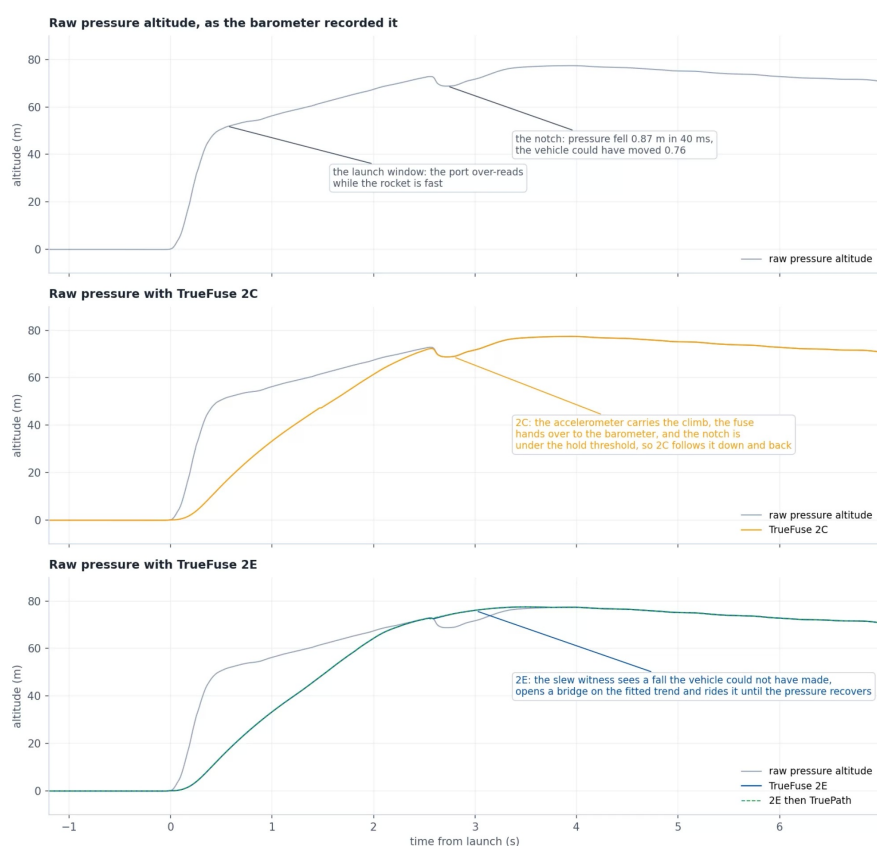
The trend sensor keeps a least-squares fit of the barometer's recent path. When the raw leaves that path by more than six metres, against both the pressure prediction and the accelerometer's, it freezes the fitted arc and holds the output on it. Before apogee is confirmed the arc ignores its newest 300 ms, as the descent always did, so an ejection spike at apogee cannot tilt it upward. While a frozen arc is climbing its curvature may not be gentler than gravity, because an unpowered climbing body

cannot decelerate more slowly than g . A hold releases once the pressure has been back across its arc for 40 ms and merges over the next 100 ms; during the merge the output must stay inside the interval between the raw and the held prediction, because a blend of two signals cannot leave its own interval.

Holds have a bounded life. If the barometer has not become believable again within the limit, the filter stops holding and returns to the barometer regardless, because a carry that runs long has no way to prove itself right. And one hold has to be refused: on an airframe whose port unwinds more than ten metres below the pad after launch, the trend sensor may not open a hold for the first three seconds of the flight, because the thing it would be holding across is the barometer coming back to the truth rather than leaving it.

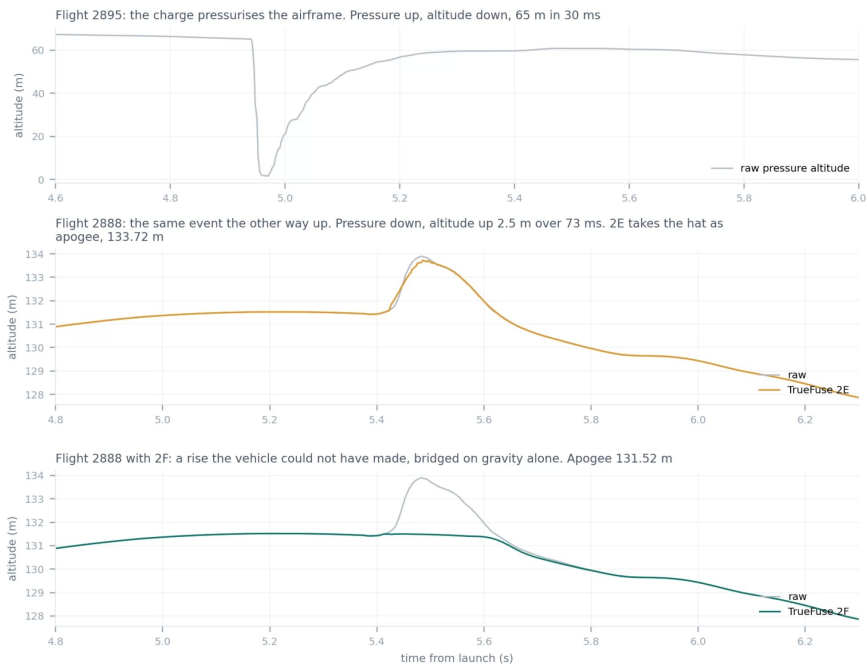
The slew witness: bridges

The hold detector cannot see small artifacts, and a four metre notch three quarters of a second before apogee is entirely capable of pulling the trace down and up again. The slew witness is the purest piece of physics in the filter: over any 40 ms the barometer may not honestly fall further than the vehicle could have moved in that time, plus the barometer's own noise as measured in the air over the previous second. When the newest sample is further below the sample 40 ms ago than that allowance, the fall did not happen, and a bridge opens on the trend the barometer was following. The bridge is a physics line, seeded at the trend's value, carried on the accelerometer's projection, capped at the height the trend's velocity could reach against gravity, and released when the raw comes back to it or apogee is confirmed.



A notch too small for the hold detector. The bridge opens the sample the fall becomes impossible, rides the fitted trend across, and releases when the pressure recovers. TruePath then back-fills the first twenty milliseconds the live filter had already let through before it could know.

The same argument runs upward. A barometer may not honestly rise faster than the vehicle can climb either, and on an airframe where the altimeter bay is sealed from the charge an ejection reads as exactly that: a smooth rise of a couple of metres over thirty samples, which no despike would ever call a glitch and which becomes the apogee if it is believed. On a rise the bridge anchors on a straight line through the 300 ms of raw before it and coasts on gravity alone, because an accelerometer inside an airframe that has just fired a charge is not a witness worth having. It releases when the raw comes back within two metres of the arc and stays there.



The same event with both signs. One flight dives sixty-five metres when the charge pressurises the airframe; the other rises two and a half metres when the pressure at the port falls instead, and that hat would have been the apogee. A hold carries the first; a gravity arc carries the second.

Where a hold and a bridge see the same event, the hold wins. The bridge's ceiling comes from a trend slope that under-reads near apogee; the hold's arc has no ceiling and releases where the pressure meets it. The bridge exists for the notches the hold detector cannot see.

Design contract

The barometer is the reference. The accelerometer never sets the altitude. It carries it between two barometric anchors, and past apogee it does not carry it at all.

No look-ahead in the live channel. Every decision uses samples already taken, and every revision is re-run with the log cut off to prove it.

Nothing is tuned to a flight. Every rule is physics or arithmetic that can be read and defended. A change that fixes the flight in front of us at the cost of three others is not a fix, and a battery of over two hundred real fleet flights, compared byte for byte, is what says so.

Declining is always available. Every guard resolves to the same safe action: stop fusing, output the barometer, say so in the header.

Sensor synchronisation

Fusing two sensors requires knowing which instant each sample describes. A barometer sample is old by the time the processor sees it: the ADC averages pressure over a conversion, the result waits to be read, and the sensor's own filter delays a ramp further. An accelerometer sample is comparatively fresh. Comparing them without correcting for this makes a perfectly good barometer look as though it disagrees with a perfectly good accelerometer, and a fusion filter fed misaligned streams produces confident nonsense.

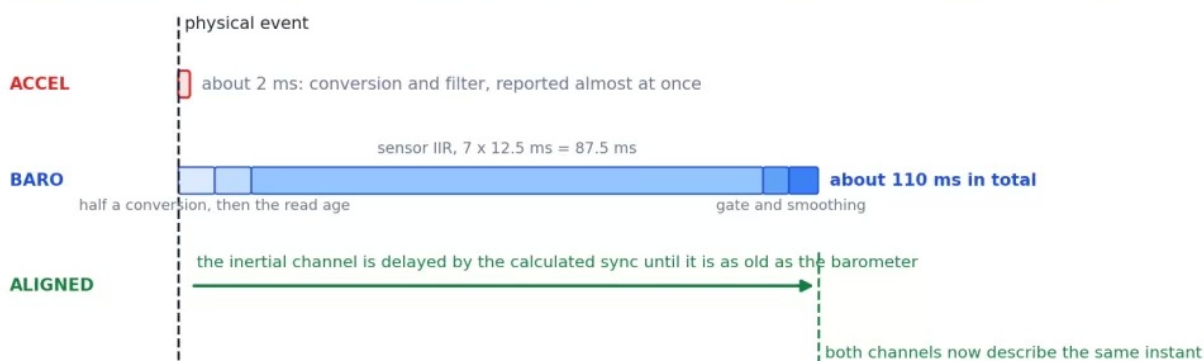
The delay is not estimated from flight data. It is a property of the configuration, so the device calculates it and aligns the streams before the filter sees a sample:

$$\text{sync} = t_{\text{conv}} / 2 + T_{\text{odr}} / 2 + c_{\text{iir}} \times T_{\text{odr}} + t_{\text{gate}} + t_{\text{kalman}} - t_{\text{imu}}$$

- `t_conv / 2` the ADC integrates over the conversion, so the value describes the middle of that interval.
- `T_odr / 2` the mean age of the most recent completed conversion when the processor takes it.
- `c_iir x T_odr` the sensor's internal IIR filter delays a ramp by exactly its coefficient in sensor samples. This is the dominant term: at a coefficient of 7 it alone is 87.5 ms on a BMP581 at 80 Hz and 140 ms on a BMP390 at 50 Hz.
- `t_gate` the extra age added by how often the device takes the barometer; on the Nano the newest frame is taken once per 100 Hz pass and every row of that pass carries it.
- `t_kalman` the altitude smoothing filter's own ramp lag, the only term that varies with the smoothing setting.
- `t_imu` the accelerometer's own conversion and filter delay, subtracted because it applies to the other side.

Where the barometer's delay comes from

BMP581 at 80 Hz, oversampling x16, IIR coefficient 7; the sync is calculated from these settings, never estimated from the flight



AltimeterCloud.com

Where the barometer's delay comes from, and why the inertial channel is delayed to match it. The sensor's own IIR filter dominates; lowering it shortens the delay and raises the noise, which is the trade the setting exists to make.

The exact figure the device used is in the header as the sync offset. The correction is carried as what it physically is, a quantity of time, never a count of samples, because a count of samples is only a time while the rate never changes. One consequence worth knowing when reading a log: every timestamp trails the physical event by the barometer's chain delay, uniformly, so intervals and rates are right even though absolute instants are late.

In depth: barometric pressure sensor lag, and how to correct it

Scale calibration

An accelerometer at rest measures exactly one gravity, so its reported magnitude at rest is a direct measurement of its own scale error. On the pad, any sample with a plausible rest magnitude and a quiet barometer seeds a running estimate; later samples join it if they sit within six percent of it, or 60 mG, whichever is larger. The calibration factor is the ratio of true gravity to the measured rest magnitude. The alternative, a fixed acceptance band around 1000 mG, fails silently on a unit whose scale is off: it admits no samples, the factor stays at one, and the filter integrates an accelerometer that is reporting the wrong number. A flight in the reference set rested at 2050 mG for exactly that reason.

What this means in practice. Nothing to configure. Leave the altimeter still for a few seconds before launch, which the pad detection needs anyway, and the calibration takes care of itself. The measured rest magnitude is in the diagnostics, so an out-of-specification sensor is visible rather than hidden.

World vertical

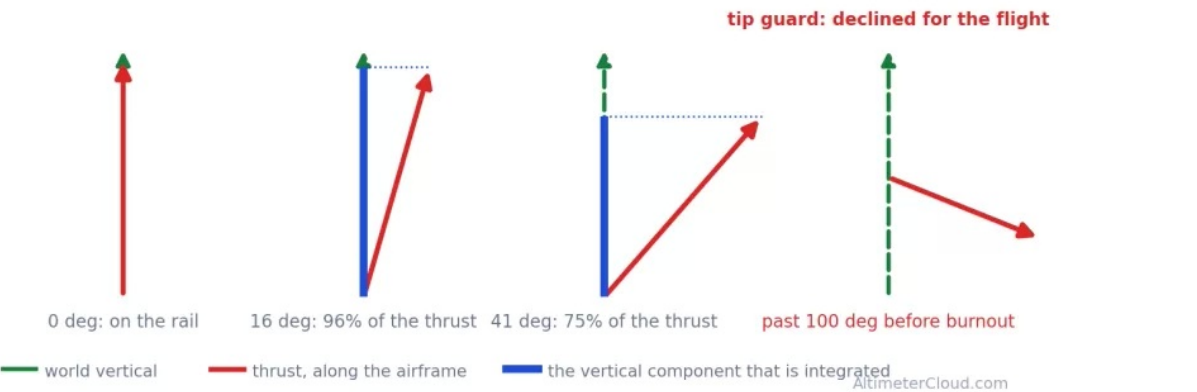
An accelerometer measures force along its own axes, which are fixed to the airframe, and rockets do not fly straight up: they tilt, weathercock and corkscrew. What gains altitude is the vertical component of thrust, not its magnitude. We tested

orientation-free magnitude early on; it is acceptable for a short bridge and measurably not good enough as a general integrand, because magnitude has no sign and rectifies noise and tumble into climb.

So the vertical is established from gravity while the vehicle is at rest and rotated with the gyroscope for the rest of the flight. Where the device's own orientation engine has proved itself, by agreeing with measured gravity at rest on the pad, the filter uses that logged attitude, with the carried gyro as the fallback.

Thrust is projected onto the world vertical

the vertical is learnt from gravity at rest and carried through every rotation; an angled flight contributes only the vertical component of its thrust



Thrust is projected onto the tracked vertical. If the vertical passes 100 degrees from the pad before burnout, the vehicle has tipped over under thrust, which no ascending rocket does, and the filter declines for the whole flight. A vertical that dies after burnout is ordinary recovery tumbling and is treated as a guard, not a verdict.

Guards

These are the conditions under which TrueFuse stops trusting the inertial sensors, and what each one does. Every one resolves to the barometer, which is why none of them can produce an inflated altitude.

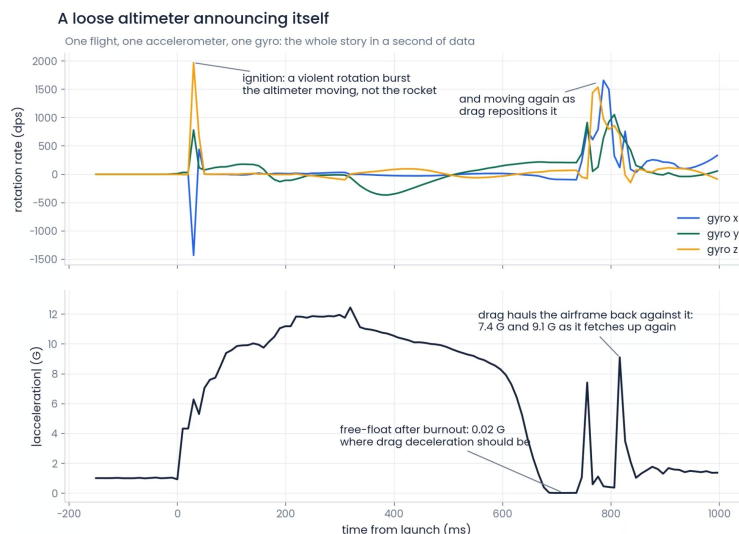
GUARD	TEST	ACTION
Tip over	tracked vertical past 100 degrees before burnout	Decline for the whole flight
Tumble	sustained end-over-end rotation the carry cannot follow	Decline from that point
Ignition slip	rotation burst at ignition, thrust line off the pad line	Magnitude bridge, re-anchor on the thrust vector
Saturation	any axis at full scale, 5 samples running	Suspend the inertial carry
Data that is not data	brown-out zeros, missing IMU rows, corrupt gyro bursts	Hold rather than believe, or decline
Hold or bridge timeout	barometer not believable again in time	Return to the barometer
False trigger	fuse fired, no launch followed	Forget it; unwind the window at save

Saturation, and why a few samples are tolerated

A clipped accelerometer reports its rail, not the real force, so the reading is a floor. Using a floor briefly understates the acceleration and pulls the fused altitude toward the barometer, which is the harmless direction. Hard motors touch full scale for a few milliseconds at ignition, and treating that as a fault would suspend the carry at the exact moment it is most needed. So saturation shorter than five consecutive samples is ignored, and anything longer is treated as a genuine out-of-range burn where integration would be dishonest. The clip threshold is derived from the configured measurement range, because the range is a setting.

A burnout stamp has to earn belief

The schedules turn on the burnout stamp, and the device's own stamp is usually right. When it is impossible, zero, after apogee or after the confirmation, a repair law rebuilds it from the accelerometer: ignition is the first two consecutive samples over 1.5 G, burnout the first three under 1.0 G after it. The candidate burn then has to have produced velocity, by the same bar the fuse uses to reject a false trigger, before it is believed, and any candidate ending after apogee is refused. That is what stops two rows of corrupt IMU data at launch detect, or a 24 G catch at the end of a hand throw, from passing themselves off as a motor. The header says when a stamp was repaired.



Why the guards exist: a loose altimeter at ignition. Top, the rotation burst as the thrust load shoves it into a seat and a second as drag repositions it after burnout. Bottom, the thrust curve, the free-float where drag deceleration should be, and the impacts as the airframe fetches up against it.

TruePath, the pass after

TrueFuse flies the flight sample by sample, never knowing the future. TruePath runs after it on every device, with the whole flight in view, and does the job the live filter cannot: it de-spikes and polishes the finished record. It recognises a gap in the log and passes the affected samples through rather than smearing across the hole, judged against the sampling rate around the gap so that hybrid mode's slower logging is not mistaken for one. A flagged run that reaches the start or end of a record is left alone rather than held flat. And it finishes the bridge's job: the live filter cannot bridge a notch before it knows the notch is there, so the first samples of every notch, typically 20 ms of them, are on the trace TrueFuse hands over. TruePath receives the bridge events, the moment the notch began and the trend it rode, and sets those samples onto the bridge's own line. Nothing of the artifact survives into the saved file. The final smooth is a nine-point median followed by a Savitzky-Golay pass, with its taps spaced so the kernel spans the same time window at every logging rate.



The division of labour: TrueFuse flies the flight sample by sample, never knowing the future; TruePath judges the finished record with both directions of time in view.

What is in your log

The recorded altitude column is the full chain: raw barometer, TrueFuse, TruePath despikes, TruePath smooth. The raw barometer is kept in its own column exactly as recorded, snapshotted before any stage runs and never written to afterwards. The saved velocity is a centred time stencil over the final altitude, a difference over a 300 ms baseline rather than a smoother, so it cannot tell a different story from the altitude it sits on. On a flight logged by firmware older than the filter, the website offers

TrueFuse and TrueFuse with TruePath as chart overlays instead, because the altitude column in those logs predates it.

The header and the flight page record what the filter did:

FIELD	MEANING
TrueFuse	The version, and whether it was applied or declined, with the reason when it declined
Trend holds	How many holds the trend sensor opened
Launch fuse windows	How many times the launch fuse carried the climb
Pad holds	Holds opened while the rocket was still on the pad
Bridge opened on a rise	Upward bridges: a rise the vehicle could not have made
Hold refused on a port dip	Whether the first-three-seconds refusal was in force, after the raw read below the pad
False-trigger windows unwound	Fuse windows rewritten back to pressure at save (Nano and website only)
Port coefficient k	The measured port error, in metres per (m/s) squared, or no correction
Largest correction	The furthest the fused trace sat from the barometer at any sample
Burnout stamp used	The stamp the schedules turned on, marked repaired if the repair law rebuilt it
Apogee stamps used	The peak and the confirmation, read from a median of the raw trace so an ejection artifact cannot contaminate them
Sensor sync	The calculated alignment applied to the inertial channel, in samples and milliseconds

A declined flight is not a failed flight. It means the filter judged the fusion unsafe and gave you the barometer, which is exactly what an altimeter without fusion would have recorded. The raw and processed apogees sit side by side on the flight page.

Validation

Every candidate change is run across a battery of real fleet flights, now over two hundred of them, and compared to the previous revision sample by sample. Not statistically: byte for byte. Because the filter is deterministic, with no covariance and no computed gain, the same inputs always produce the same output to the last bit, so a regression cannot hide in noise. Several mechanisms described in the articles were built, run, measured and taken out again by exactly this process, and the code carries the flight number that killed each one.

Then there is the tape measure. Thrown from the floor of the office to a ceiling at 3.10 m, the fusion came out at 3.066 m where the pressure altitude alone gave 2.767 m, despite the altimeter spinning in the air and being thrown by hand. And the first real flight through the bridge, a small B-motor flight saved by a Nano, had its sixty-three metre ejection dive carried across by the trend sensor and came back to the reference within sixteen millimetres.

Please note that this test throw did invent the true apogee as it was permitted to do so for testing. On a real flight this is not possible, nor would it occur vs a 3 meter throw in an office.



When the same law that bridges three-hundred-metre ejection spikes also wins an argument with a tape measure, we call that a good week.

Honest by design

The .aclz log always keeps the unfiltered pressure altitude in its own column, next to the fused trace. The header states the filter version, whether it ran or declined and why, which stamps it used and whether one was repaired, and the largest correction it made. The fused velocity is derived from the fused altitude by a plain time stencil, so it cannot tell a different story than the altitude it sits on. You never have to take the filter's word for anything: the evidence to judge it ships inside every flight. That is not a feature we added to the filter. It is the reason the filter deserves to exist.

TrueFuse 2F revision 3 and TruePath revision 6 are what the website runs and what current Nano, Mercury and Jupiter firmware carries.