

N997CZ — Airspeed Calibration Summary

Living document. Generated 2026-05-20 18:51 UTC. Source: *AirspeedCalibration_Summary.md*.

Pitot-static / airspeed-system calibration for N997CZ (RV-10, Garmin G3X). Each new cal flight adds a row to the data table and may shift the conclusions.

1. What we're measuring

The airspeed indicator displays **IAS** (Indicated Airspeed) — the raw pitot-static reading. To turn that into something useful you walk two corrections:

$$\text{IAS} \xrightarrow{\text{position-error correction}} \text{CAS} \xrightarrow{\text{density correction (OAT + PA)}} \text{TAS}$$

IAS → **CAS** removes the installation/position error of the pitot-static system on this specific airframe. *This is what we want to calibrate.*

CAS → **TAS** is pure physics (density ratio $\sigma = \delta \cdot T_{\text{std}} / T_{\text{actual}}$), exact if OAT and pressure altitude are correct.

The airspeed-cal flight uses **GPS wind solves** to recover TAS as ground truth (it falls out of the geometry — no avionics assumptions). Then we work backwards:

$$\text{TAS}_{\text{GPS}} \times \sqrt{\sigma} \rightarrow \text{CAS}_{\text{truth}} \rightarrow \text{position error} = \text{IAS} - \text{CAS}_{\text{truth}}$$

Position error tabulated vs IAS is the airspeed-system calibration.

2. Methods

2a. Constant-IAS GPS circle (F7 method)

Fly a coordinated turn at fixed IAS and altitude. Plot the (V_E, V_N) ground-velocity samples in 2D. They trace a **circle**:

- **Radius = TAS** (constant — CAS fixed, alt/temp don't change ⇒ TAS fixed).
- **Center = wind vector** (constant).

Algebraic (Kasa) least-squares circle fit recovers both. Heading-independent — needs zero compass calibration, only GPS velocity.

Pros: clean, single-parameter (TAS), robust, immune to compass error. **Cons:** needs full 360° track coverage; only one IAS per circle. Scripts: `f7_circle_solve.py`, `f9_circle_solve.py`.

2b. Multi-leg wind triangle (F9 method, "box")

Fly $N \geq 4$ stabilized straight-and-level legs on different headings at the same IAS/altitude. For each leg, ground velocity = $\text{TAS} \times \text{unit}(\text{true_heading}) + \text{wind}$:

$$\begin{aligned} V_{E,i} &= \text{TAS} \cdot \sin(\text{true_hdg}_i) + W_E \\ V_{N,i} &= \text{TAS} \cdot \cos(\text{true_hdg}_i) + W_N \end{aligned}$$

Three unknowns (TAS, W_E , W_N), 2N equations ⇒ least-squares overdetermined fit when $N \geq 2$ (in practice want ≥ 4 well-spread headings for confidence).

Pros: works with discrete legs; AP-flown legs produce extremely tight fits. **Cons:** uses magnetic heading — vulnerable to compass / variation sign errors (see §5). Scripts: `f7_box_solve.py`, `f9_box_solve.py`.

2c. Leg detection

Both methods need stabilized data. `f7_lettered_legs.py` / `f9_lettered_legs.py` auto-detect windows where heading, IAS, altitude, and (optionally) bank angle stay inside tight tolerances for $\geq N$ seconds.

3. Data points collected to date

3a. Cal flights (geometry + atmosphere + wind solve)

#	Flt	Date	Method	KIAS	PA (ft)	OAT (°C)	σ	TAS_GPS (kt)	Wind	Fit RMS
---	-----	------	--------	------	---------	----------	----------	--------------	------	---------

1	F7	2026-05-15	Const-IAS circle (1,336 samples, 358° track cov.)	147.9	6,927	+4	0.766	168.1	23 kt FROM 341°T	2.4 kt
2	F9	2026-05-17	11-leg AP box + GPS circle	144.7	8,316	+8	0.709	171.3	28 kt FROM 266°T	1.6 (circle) 2.0 (box)

3b. Derived errors (the calibration result)

Flt	IAS	CAS_truth (=TAS·√σ)	TAS_GPS (truth)	TAS_G3X (logged)	IAS – CAS (position error)	TAS_G3X – TAS_GPS (G3X bias)
F7	147.9	147.1	168.1	164.7	+0.9 kt	–3.4 kt
F9	144.7	144.2	171.3	166.6	+0.5 kt	–4.6 kt

All values in knots. Highlighted columns are the calibration result and the G3X-internal bias finding.

4. What we’ve learned

4a. The pitot-static system is essentially clean in cruise

At ~145–148 KIAS, position error is **+0.5 to +0.9 kt** — within the ±2 kt uncertainty of the GPS-circle method itself. Two independent days, two different winds (341°T vs 266°T), two different solver methods, two different altitudes (6,900 vs 8,300 ft). All converge on “essentially zero position error in cruise.” That’s a much stronger statement than either flight alone.

4b. The G3X-computed TAS reads ~3–5 kt LOW

A separate bias. F7: G3X logged TAS averaged **3.4 kt below** GPS truth. F9: **4.6 kt below**. If the G3X simply did $TAS = IAS / \sqrt{\sigma}$ with the same sensors we’re using, the G3X-TAS error and the position error would scale together. They don’t — the indicator is nearly perfect but the G3X’s TAS is ~4 kt low. Possible causes, ranked by likelihood:

1. **G3X has an internal IAS→CAS table** subtracting ~3–4 kt as “installation correction” by default. Since the actual position error is ~0, that internal correction is *overcorrecting* and dragging G3X’s TAS too low. Configurable on the G3X airspeed-cal page — worth comparing current entries to the measured GPS-truth.
2. OAT-probe bias. Quantitatively too small: 3°C error → ~0.8 kt TAS shift. Not enough.
3. Compressibility correction at $M \approx 0.27$. Sub-1 kt. Not it.

Net: the G3X’s internal calibration table is the prime suspect. Indicator itself is fine. Action: pull up the G3X airspeed-cal page on the ground, compare entries against the actual measured position error in this document, zero out any spurious entries.

4c. Wind solves cross-validate themselves

In F7, the constant-IAS circle gave RMS 2.4 kt. In F9, the GPS circle gave RMS 1.6 kt and the 11-leg wind-triangle gave RMS 2.0 kt — and the two F9 methods agreed to **0.02 kt** on TAS and within 3° on wind direction. End-to-end methodology check: when the data and the math are right, the two methods *must* agree.

4d. AP-flown legs are dramatically cleaner than hand-flown

F9’s autopilot tuning flight produced 11 wings-level ($|\text{roll}| \leq 3^\circ$), constant-IAS (± 0.7 kt) legs across all four cardinals without anyone trying. The F7 hand-flown circle worked but accumulated more noise, and the 4-leg discrete box attempt inside F7 was poor enough to be initially misdiagnosed as a bank-angle problem (see §5). **Takeaway:** schedule the next multi-speed cal as an AP mission, not hand-flown.

5. Methodology lessons

5a. Magnetic-variation sign convention (the bug)

`f7_box_solve.py / f9_box_solve.py` originally computed $\text{true_hdg} = \text{mag_hdg} + \text{mag_var}$. **Wrong for this dataset.** At KHEF actual declination is ~10° West; the G3X logs Magnetic Variation as **+10.1°** (its column is West-positive, not East-positive). The right formula is $\text{true_hdg} = \text{mag_hdg} - \text{mag_var}$.

With the wrong sign, every leg in the F9 box solve showed a track-minus-heading drift of -11° to -30° — physically impossible for a single wind. *That* was the diagnostic. Flipping the sign dropped both flights' box-fit RMS from ~ 11 kt to ~ 2 kt and produced answers that match the heading-independent circle method to 0.02 kt TAS.

Earlier diagnosis of F7's box-solve failure ("legs flown banked/turning") was **incorrect** — the legs were fine, the script was 20° off in true heading. F9 (AP-flown wings-level) reproduced the same RMS-12 failure under conditions where bank-angle couldn't have been the cause, which is what pinned the bug on the formula.

5b. Prefer the GPS-circle method when the compass is in question

If you ever see all-same-sign residuals in the box solve, run the circle solve as a sanity check. The circle method doesn't touch the compass — if its TAS disagrees with the box solve's TAS, the compass / variation / heading interpretation is the suspect, not the airspeed system.

5c. AP makes everything easier

For deliberate cal flights: engage AP, fly a racetrack pattern at one target IAS / altitude, capture 4+ stabilized legs at distinct cardinal headings, then turn 90° and do another lap. F9 did this by accident during AP tuning and produced the cleanest cal data we have.

6. What's still missing — open work

6a. Multi-speed coverage

Both cal points are at ~ 145 – 148 KIAS. Position error is **often non-linear** — a system reading $+0.5$ kt at 145 KIAS could easily read $+3$ to $+5$ kt at 70 KIAS. **Needed:** cal points at additional IAS values across the envelope. Reasonable plan: 80, 100, 120, 140, 160 KIAS, each as a single AP-stabilized box (4 legs at cardinal headings, ~ 20 s each) at constant altitude (say 6,000 ft to keep σ comparable). Total flight time ≈ 25 – 40 min if turns are tight.

6b. G3X airspeed-cal table

Check whether the G3X has a configurable IAS-to-CAS table loaded and whether its entries match the measured zero position error. If a non-zero correction is loaded by default, it's the likely source of the -3 to -5 kt G3X-TAS bias.

6c. OAT-probe sanity check

While the math says OAT alone can't explain the G3X-TAS bias, sanity-checking the OAT probe against an independent thermometer on the ground (or against ATIS at startup) is cheap insurance.

7. Reference — scripts in [Analysis/](#)

Script	Purpose
f7_lettered_legs.py	Auto-detect stabilized legs in F7, gated to $\sim 7,000$ ft / ~ 150 kt cal block
f7_legs_by_roll.py	Same but tracks bank angle per leg (used during banked-leg misdiagnosis)
f7_box_solve.py	4-leg wind-triangle on F7 D–G legs (patched: true = mag – var)
f7_circle_solve.py	Algebraic Kasa circle fit on F7 continuous turn — heading-independent
f9_lettered_legs.py	Auto-detect stabilized AP legs in F9 ($ \Delta\text{hdg} \leq 6^\circ$, $ \Delta\text{IAS} \leq 1.5\text{kt}$, $ \Delta\text{alt} \leq 50\text{ft}$, $ \text{roll} \leq 3^\circ$)
f9_box_solve.py	11-leg wind-triangle on F9 stabilized legs (patched: true = mag – var)
f9_circle_solve.py	Kasa circle fit on the same 11 F9 leg medians
make_airspeed_cal_pdf.py	Renders this PDF from the same content as AirspeedCalibration_Summary.md

8. Change log

- **2026-05-20** — F9 cal added ($+0.5$ kt position error at 145 KIAS / 8,300 ft). Discovered & fixed the `mag_var` sign bug in both box-solve scripts; corrected the F7 box-solve "banked-legs" misdiagnosis. Cross-validated F7 and F9 against the GPS-circle method (heading-independent).
- **2026-05-18** — F7 cal recorded ($+0.9$ kt position error at 148 KIAS / 7,000 ft, GPS circle method only — box solve was thought to be invalid).