Glossary

A working reference for satellites, measurements, formats, and the terms specific to this repo.

---

1. Satellites and signals

- **GNSS** — Global Navigation Satellite System. Umbrella term for all constellations.
- **Constellation** — one operator's satellite fleet:
  - GPS (USA, sat ids `G01…`)
  - GLONASS (Russia, `R`)
  - Galileo (EU, `E`)
  - BeiDou (China, `C`)
  - QZSS (Japan, `J`)
  - NavIC (India, `I`)
  - SBAS (augmentation, `S`)

  The ZIM3 file has 55 sats because it tracks all of them.
- **PRN / sat id** — satellite number inside a constellation. `G04` = GPS satellite 4.
- **L1 / L2 / L5 / E5 / B2** — radio frequencies (bands). L1 ≈ 1575.42 MHz, L2 ≈ 1227.60 MHz, L5 ≈ 1176.45 MHz. Multi-frequency helps because ionosphere delay depends on frequency, so two bands let you cancel it.
- **Signal** — a band plus a modulation code, e.g. `GPS_L1CA` = GPS L1 Coarse/Acquisition. One satellite emits several signals.

---

2. What the receiver actually measures

- **Observation (obs)** — one measurement of one signal from one satellite at one time.
- **Epoch** — one time instant containing all observations. These files have one epoch every 30 s (ZIM) or 15 s (ab02).
- **Pseudorange** — travel-time distance satellite-to-receiver, in metres. Called *pseudo* because receiver clock error is baked in, so it's off by a common bias, typically hundreds of km. Noise ~0.3–3 m. Fields `C1C`, `P2`, `C1` in RINEX.
- **Carrier phase** — distance measured by counting carrier wave cycles. Noise ~2 mm, 1000× better than pseudorange. Problem: you only know the fractional part; the integer

number of whole cycles is unknown. That unknown is the **ambiguity**. Fields `L1`, `L2`.
- **Doppler** — frequency shift, gives velocity directly.
- **SNR / CN0** — signal strength in dB-Hz. ~45 = strong, <30 = weak or reflected. Fields `S1`, `S2`.
- **Cycle slip** — receiver loses phase lock for an instant, integer ambiguity jumps. Must be detected or position breaks.
- **LOS / NLOS** — Line Of Sight / Non-Line Of Sight. LOS = direct path sat-to-antenna. NLOS = direct path blocked, only a reflected signal arrives, which is longer, so range is biased metres to tens of metres too long. Main urban error source, and exactly what the robot-body mask kills.
- **Multipath** — direct signal plus reflection arrive together, interfering. Related to NLOS but not identical.

---

3. Error sources

- **Ionosphere** — charged upper atmosphere, delays code / advances phase, 1–30 m, frequency-dependent.
- **Troposphere** — lower wet/dry atmosphere, ~2.3 m at zenith, grows as `1/sin(elevation)`. Not frequency-dependent, so must be modelled (Saastamoinen) or estimated.
- **IFLC / ionosphere-free combination** — arithmetic mix of two frequencies that cancels ionosphere. Costs noise.
- **Ephemeris** — the orbit model of a satellite. Needed to know where the sat was when it transmitted.
- **Broadcast ephemeris** — orbit data transmitted by the satellites themselves. Accuracy ~1–2 m. This is the `brdc` / `.nav` / `MN.rnx` file.
- **Precise products** — computed after the fact by analysis centres:
  - `SP3` = precise orbits (~2.5 cm)
  - `CLK` = precise satellite clocks
  - `IONEX` = global ionosphere maps
  - `DCB` / `OSB` / Bias-SINEX = hardware code biases per satellite/signal
- **SSR** — State Space Representation. Orbit/clock/bias corrections streamed in real time instead of computed later. CLAS and MADOCA are Japan's QZSS SSR services, broadcast on the L6 signal.

---

4. File formats

RINEX — Receiver INdependent EXchange format. Plain text, the universal GNSS interchange format. Two kinds matter here:

- observation file — the measurements
- navigation file — the ephemerides

RINEX 2 vs RINEX 3 — version 2 is old, cramped layout, one global list of observation types. Version 3 is modern, per-constellation observation types, better multi-GNSS. The parser bug hit above is in the version 2 path only.

File naming, old (RINEX 2) — `ssssDDD0.YYt`

Example	Meaning
`ab022370.26o`	station `ab02`, day-of-year 237, session 0, year 26, type `o` = observation
`brdc2370.26n`	broadcast, DOY 237, year 26, `n` = GPS navigation
`.26g` / `.26p`	GLONASS nav / mixed nav

File naming, new (RINEX 3) — long form

`ZIM300CHE_R_20262370000_01D_30S_MO.rnx`

Field	Meaning
`ZIM3`	station
`00`	monument/receiver number
`CHE`	country (Switzerland)
`R`	from data centre archive
`2026 237 0000`	start year, DOY, hour+minute
`01D`	1 day span
`30S`	30 second sampling
`MO`	Mixed constellation Observation (`MN` = Mixed Navigation)
`.rnx`	plain RINEX text

Other formats

- **Hatanaka / CRINEX / `.crx`** — RINEX observation compressed by a GNSS-specific algorithm (stores differences between epochs). Halves size before gzip. Must decompress with `crx2rnx` or Python `hatanaka` before this library reads it. `.crx.gz` = Hatanaka then gzip.

- `.pos` — solution output. One row per epoch: week, tow, ECEF XYZ, lat/lon/height, quality flag, satellite count, DOP, extra diagnostics. Convention from RTKLIB.
- **Also in this lib** — RTCM (binary correction/observation streaming standard, used by NTRIP casters), UBX (u-blox receiver binary), SBF (Septentrio binary), NMEA (old ASCII sentences, `$GPGGA…`), BINEX (binary exchange), KML (Google Earth track).

---

5. Time

- **GPS week** — weeks since 1980-01-06. This data: 2433.
- **tow** — Time Of Week, seconds since Sunday 00:00 GPS time, range 0…604800. `172800.000` = 48 h into the week = Tuesday 00:00. Week + tow uniquely stamps an epoch. It's the join key in the `los_prob` CSV because it's what the solver holds internally.
- **Leap seconds** — GPS time does not have them, UTC does; currently 18 s apart.

---

6. Coordinate frames

- **ECEF** — Earth-Centred Earth-Fixed. XYZ metres, origin at Earth's centre, rotates with Earth. What the solver works in. Zimmerwald ≈ `(4331300, 567537, 4633133)`.
- **Geodetic / LLH** — latitude, longitude, height above the reference ellipsoid WGS84. Same point, human-readable.
- **Ellipsoidal vs orthometric height** — ellipsoid height is geometric; orthometric is height above the geoid (mean sea level). Differ by tens of metres.
- **ENU** — East, North, Up. Local flat-earth frame tangent at your position. Good for "how far did I move" and for sky directions.
- **NED** — North, East, Down. Same idea, aviation convention. Library exposes velocity in both.
- **Azimuth** — compass direction to a satellite, degrees, 0 = North, increasing clockwise (90 = East).
- **Elevation** — angle above local horizon, 0 = horizon, 90 = straight up (zenith).
- **Elevation mask** — reject satellites below N degrees. Low satellites travel through more atmosphere and reflect off everything. Default 15° here.
- **Attitude** — orientation of your robot: roll (tilt sideways about forward axis), pitch (nose up/down), yaw (heading). ZYX means apply yaw, then pitch, then roll.
- **REP-103 / FLU** — ROS convention: body x Forward, y Left, z Up; world frame ENU. What `body_mask.py` uses.
- **Quaternion** — 4-number rotation representation, no gimbal lock. `include/libgnss++/fusion/attitude.hpp` uses these.

---

7. Positioning methods

Method	Description	Accuracy	Binary
SPP	Single Point Positioning. One receiver, pseudoranges only, broadcast ephemeris. Fast, always works, no base station.	1.5–10 m	`gnss_spp`
DGNSS	Differential; base station sends corrections, code only.	~0.5–2 m	—
RTK	Real Time Kinematic. Two receivers, carrier phase, resolve the integer ambiguities. Needs a base within roughly 10–40 km.	1–2 cm when fixed	`gnss_solve`
PPP	Precise Point Positioning. Single receiver, but precise orbit/clock products instead of a base. Takes 10–30 min to converge.	Few cm	`gnss_ppp`
PPP-RTK / CLAS / MADOCA	PPP plus atmospheric corrections from a network, so it converges fast like RTK without a nearby base.	—	—

---

8. RTK vocabulary

- **Base / rover** — base is the station on a known point, rover is the one you're positioning. In this test ZIM2 = base, ZIM3 = rover.
- **Baseline** — vector (and distance) base-to-rover. Shorter = better; errors cancel because both antennas see the same atmosphere. The Zimmerwald pair is co-located, so essentially the easiest possible case, hence 96.67% fix.
- **Single difference (SD)** — rover measurement minus base measurement to the same satellite. Cancels satellite clock and most atmosphere.
- **Double difference (DD)** — difference of two single differences, using a reference satellite. Also cancels receiver clocks and leaves an integer-valued ambiguity. This is the core of RTK, and it's why the exclusion guard protects "a system's last reference satellite" — remove it and that whole constellation's DD set collapses.
- **Ambiguity resolution (AR)** — determining the integer cycle counts. LAMBDA is the standard search algorithm.

- **Float solution** — ambiguities left as real numbers. Decimetre level. Quality flag `Q=2` in `.pos` (here 3 also appears for other modes).
- **Fixed solution** — ambiguities resolved to integers. Centimetre level. `Q=1` classic RTKLIB, `Q=4` in this library's output.
- **Ratio test** — best-vs-second-best candidate ratio; above threshold (default ~3) the fix is accepted.
- **Fix rate** — fraction of epochs that reached a fixed solution. Baseline here: 96.67%.
- **Wrong fix** — fixed integers that are wrong. Dangerous: confidently reports centimetres while being metres off. The README tables track it separately for that reason.
- **Fix-and-hold** — once fixed, constrain the ambiguities to keep them.

---

9. Filters and estimation

- **Least squares / WLS** — solve for position from redundant measurements; the weighted version trusts high-elevation, high-SNR measurements more.
- **Kalman filter (KF)** — recursive estimator carrying position and ambiguities forward through time with a covariance.
- **FGO** — Factor Graph Optimization. Instead of one forward pass, optimise a window of recent epochs jointly. Better in bad environments, more compute. GTSAM is the library used. `gnss_fgo`.
- **Tightly coupled GNSS/IMU** — fuse raw satellite measurements with the IMU inside one estimator, rather than fusing final positions. Survives short outages.
- **RAIM / FDE** — Receiver Autonomous Integrity Monitoring / Fault Detection and Exclusion. Leave one satellite out at a time, see if residuals improve; if so, that satellite is faulty. Automatic outlier removal.
- **Residual** — measured minus predicted. Small residuals = consistent solution.
- **Robust / Huber weighting** — downweight rather than delete large residuals.

---

10. Quality numbers

- **DOP** — Dilution Of Precision. Pure geometry factor: how the satellite spread amplifies measurement error. Lower = better. GDOP geometric, PDOP position, HDOP horizontal, VDOP vertical. Under 2 = excellent, over 10 = poor. The 40° mask run hit GDOP 29 because only 4 satellites clustered overhead remained — exactly the failure mode aggressive robot masking can cause.
- **Availability** — fraction of epochs with any valid solution.
- **RMS** — root-mean-square error against truth.
- **P95** — 95th-percentile error. Says what your bad epochs look like, unlike RMS which averages them away.

- **pp** — percentage points (difference of two percentages).

---

11. Networks, stations, archives

- **CORS** — Continuously Operating Reference Stations, the US NOAA/NGS network. Free, open HTTP, no login. Source of `ab02`, `p091`.
- **IGS** — International GNSS Service. Global scientific network, produces the precise SP3/CLK products.
- **EUREF** — the European reference network.
- **BKG** — German federal agency running an open IGS/EUREF data mirror. Where the Zimmerwald files came from.
- **Zimmerwald** — an observatory near Bern, Switzerland. ZIM2 and ZIM3 are two receivers at that same site, a few metres apart, which is why RTK fixes almost every epoch. Good property for testing: any degradation you see is caused by your mask, not by the baseline.
- **brdc** — "broadcast", the daily merged broadcast-ephemeris file.
- **NTRIP** — protocol for streaming RTCM corrections over HTTP. Caster = the server. Live RTK path, `gnss_stream` / `gnss_live`.
- **DOY** — day of year, 001–366. 237 = 2026-08-25.

Revision #3
Created 2026-08-31 19:21:46 UTC by Jakub Kecek
Updated 2026-08-31 19:28:51 UTC by Jakub Kecek