EZRA ROBOTICS

X30 Tuning Guide

The models behind every setting of the X30 stack, how to measure them on the robot, and the complete parameter reference

Version 1.0 • 2026-11-21

For the engineers who commission and tune the X30: the braking model and the protective field, the scans, odometry and localization, route following and avoidance, the inspection criteria. Basic ROS knowledge assumed.

About this document

For whom. For the engineers who commission and tune the X30: the braking model and the protective field, the scans, odometry and localization, route following and avoidance, the inspection criteria. Basic ROS knowledge assumed.

Installing the stack is in the X30 Installation Guide, operating it in the X30 User Manual, the nodes, topics and protocols in the X30 Architecture & Technical Reference. The parameter tables in Appendix A are generated from the sources (`docs/tools/params.py`) and always match the code of this version.

The X30 documents

Installation Guide	Getting the stack, the handheld app and NETRA onto a robot and a computer, from nothing to the first patrol.
User Manual	Operating the robot: safety, the Skydroid H12, routes, patrols, inspections, the recorder and NETRA.
Tuning Guide	What every setting does, the models behind them and how to tune a robot for a site; the full parameter reference.
Architecture	How the stack is built: hosts, nodes, topics, frames, the safety pipeline, the protocols to the handheld and to NETRA.
Site Acceptance Test	The handover tests, with pass criteria and sign-off.

All of them, and the same pages as a website with search: `https://darmawanrobotics.github.io/x30_ros/`.

Revisions

Version	Date	Changes
1.0	2026-11-21	First version, for x30_ros 0.19

Contents

About this document . . . 2

1 How the settings are organized . . . 5
- 1.1 Five places . . . 5
- 1.2 Live and start-only parameters . . . 5
- 1.3 Where to start . . . 5

2 The robot profile and its braking model . . . 7
- 2.1 The profile . . . 7
- 2.2 One braking model . . . 7
- 2.3 Measuring the braking . . . 8
- 2.4 The base height . . . 8

3 The driver: speeds to the robot . . . 9
- 3.1 From a velocity to an axis command . . . 9
- 3.2 The signs of the axes . . . 9
- 3.3 Silence and stopping . . . 10

4 The control mux and the protective field . . . 11
- 4.1 Sources, silence and the e-stop . . . 11
- 4.2 The envelope . . . 11
- 4.3 The protective field . . . 12
- 4.4 Tuning the field . . . 13

5 Perception: the sector scans . . . 14
- 5.1 Which points are obstacles . . . 14
- 5.2 A blind LiDAR . . . 14
- 5.3 The near-field blind zone . . . 15
- 5.4 Motion smear . . . 15
- 5.5 Tuning the band . . . 15

6 Odometry . . . 16
- 6.1 The start gate . . . 16
- 6.2 The monitor . . . 16
- 6.3 The trip meter . . . 17

7 Localization . . . 18
- 7.1 No map: the start spot and the tags . . . 18
- 7.2 The filter . . . 18
- 7.3 The states . . . 19
- 7.4 Tags . . . 19

8 Following a route . . . 21
- 8.1 From a recording to a route . . . 21
- 8.2 The speed profile . . . 21
- 8.3 The path controllers . . . 22
- 8.4 Stops, joining and dwelling . . . 22

8.5 Smoothness . . . 23
8.6 The supervisor . . . 23

9 Getting past obstacles . . . 24
9.1 Path shift . . . 24
9.2 Detours and backing up . . . 24
9.3 Reactive avoidance . . . 25
9.4 The guard . . . 25

10 Inspection and recording . . . 26
10.1 At a stop point . . . 26
10.2 Thermal criteria . . . 26
10.3 Partial discharge . . . 27
10.4 Reports . . . 28
10.5 The recorder . . . 28

11 Health, the handheld and NETRA . . . 29
11.1 The health monitor . . . 29
11.2 The handheld gateway . . . 29
11.3 The NETRA uplink . . . 30
11.4 The simulator . . . 30

A Parameter reference . . . 31
A.1 Robot profile . . . 31
A.2 Driver . . . 31
A.3 Control mux and protective field . . . 33
A.4 Perception . . . 34
A.5 Odometry . . . 37
A.6 Localization . . . 38
A.7 Navigation . . . 40
A.8 Path controllers . . . 45
A.9 Reactive avoidance . . . 48
A.10 Docking . . . 49
A.11 Health . . . 51
A.12 Inspection . . . 52
A.13 Recorder . . . 54
A.14 Handheld gateway . . . 55
A.15 NETRA uplink . . . 56
A.16 Simulator . . . 57

CHAPTER 1

How the settings are organized

1.1 Five places

Place	File	What goes there
Robot profile	`x30_bringup/config/robot.yaml`	the body and its motion limits, read by every node under `robot/`; read-only at runtime
Node settings	`x30_<package>/config/*.yaml`	each node's own parameters (Appendix A)
Launch arguments	`/etc/x30/<role>.env`, `X30_LAUNCH_ARGS`	which parts run on a host and with what: the site, the controller, NETRA, the camera hosts
Secrets	`/etc/x30/<role>.env`	tokens and camera passwords, as environment variables only (never parameters: `roslaunch` prints parameters)
Sites	`~/x30/map/<site>/`	routes, stop points, tags and inspection points of a substation: taught, not tuned

The profile is the one place for numbers several nodes must agree on: the footprint, the speeds, the braking. A node never keeps its own copy of them; where a node needs a stricter value (the follower's cruise speed below `max_vx`), its own parameter can only lower it.

1.2 Live and start-only parameters

Every parameter is declared in a configuration dataclass with its unit, range and description, and validated when the node starts: a value outside its range or a contradiction (an approach speed above the cruise speed) stops the node with the reason. The tables of Appendix A come from the same declarations.

- A **live** parameter is read again every second. Changing it on the parameter server takes effect within that second; an invalid value is rejected with a log line and the old value stays.
- A parameter marked † is read at start only: change it in the YAML file and restart the host's service.

```
rosparam set /control_mux_node/field/margin 0.2
rosparam get /control_mux_node/field
```

The first is live, in effect within 1 s; the second shows what the node runs with.

> **A live change does not survive a restart** The parameter server is filled from the YAML files at every start. Try a value live, then write it into the YAML file of the workspace (or the package's) so the next start keeps it.

1.3 Where to start

Tune from the bottom up: each layer relies on the one below being right.

1. **The profile**: measure the base height, the braking and the reaction time (Chapter 2). Every safety distance follows from them.
2. **The driver**: the signs of the axes and the gait's speed limits (Chapter 3).
3. **The scans**: the LiDAR's mount and the height band; a scan that misses an obstacle or sees the floor makes every layer above wrong (Chapter 5).
4. **The protective field**: its margins on the measured braking (Chapter 4).
5. **Odometry and localization**: the drift and the tags (Chapters 6 and 7).
6. **Route following** and **avoidance** last (Chapters 8 and 9).

> **Record before and after** A change is judged on data, not on a feeling: `rosbag record` the topics of the layer before and after the change, on the same route, and compare. The trip meter (`odometry/trip`) and `navigation/status` give the time per lap and the stops.

CHAPTER 2

The robot profile and its braking model

2.1 The profile

`x30_bringup/config/robot.yaml` describes the X30 once for the whole stack. Its values come from Deep Robotics' manuals and from measurements on the robot:

Parameter	X30	Where it comes from
`kinematics`	`omni`	legged: forward, sideways and turning at once
`footprint.front`, `rear`	0.50 m	standing length 1 000 mm, base_link in the middle of the body
`footprint.half_width`	0.235 m	standing width 470 mm
`base_height`	0.48 m	base_link above the floor when standing: **measure**
`limits.max_vx`	0.75 m/s	Walk gait: 1.2 m/s forward, 0.75 m/s backward
`limits.max_vy`	0.15 m/s	Walk gait sideways (L-Walk 0.5 m/s)
`limits.max_wz`	0.45 rad/s	Walk gait (L-Walk 1.2 rad/s)
`limits.max_accel`	0.5 m/s^2	comfortable start
`limits.max_decel`	0.6 m/s^2	braking: **measure** (Section 2.3)
`limits.max_yaw_accel`	1.5 rad/s^2	yaw braking: **measure**
`limits.reaction_time`	0.3 s	a 10 Hz LiDAR, the scan, the mux, the 50 Hz axis stream

The limits are what the stack may command, so every gait the robot walks in must reach them: the Walk gait's 0.15 m/s sideways is why `max_vy` is low. The profile is read-only at runtime; a change needs a restart of every host.

2.2 One braking model

Every speed limit of the stack (the protective field, the follower's guard, the speed profile before a stop point, path shift) uses one model of how far the robot travels from the moment a command at speed v is sent until it stands:

$$d(v) = v\,t_r + \frac{v^2}{2a} \tag{2.1}$$

with t_r the reaction time (`reaction_time`: sensing, transport, the robot's own latency) and a the braking deceleration (`max_decel`). Read backwards, it gives the highest speed that still stops within a free distance D:

$$v_{\max}(D) = a\left(-t_r + \sqrt{t_r^2 + \frac{2D}{a}}\right) \tag{2.2}$$

The same two equations hold for turning, with the yaw rate ω, the yaw braking α (`max_yaw_accel`) and the free angle θ instead: $\theta(\omega) = \omega t_r + \omega^2/2\alpha$. At 0.45 rad/s the X30 turns another 0.20 rad (11.6°) before it stands.

> **Why the reaction time matters most at low speed** At 0.25 m/s the reaction term $v\,t_r$ is 7.5 cm of the 12.7 cm the robot needs to stop; at 0.75 m/s it is 22.5 cm of 69 cm. An underestimated t_r shortens every distance by $v\,\Delta t_r$, an underestimated a by a share that grows with v^2.

2.3 Measuring the braking

Measure on the substation's own ground, with the payload mounted and a charged battery: a heavier robot or a slippery surface brakes later.

1. Mark a straight, free 10 m lane. Record the command and the odometry:

   ```
   rosbag record -O brake.bag /x30/cmd_vel /Odometry
   ```

2. In REMOTE, walk the robot along the lane at a steady speed v for at least 3 m, then let go of the stick at once (the command drops to zero).
3. Repeat three times each at 0.25, 0.5 and 0.75 m/s.
4. From the bag, for each stop: t_0 the time the command dropped, t_1 the time the odometry speed starts to fall, t_2 the time it reaches zero, and the distance d travelled from t_0 to t_2.

Then $t_r = t_1 - t_0$, and from the model

$$a = \frac{v^2}{2\,(d - v\,t_r)} \tag{2.3}$$

For example, $v = 0.5$ m/s stopping in $d = 0.42$ m with $t_r = 0.3$ s gives $a = 0.46$ m/s^2. Put in the profile the **largest** t_r and the **smallest** a of all runs: the model must not stop shorter than the robot.

> **Never round the measurements toward the robot's favour** A higher `max_decel` or a lower `reaction_time` than measured makes every safety distance of the stack too short. When in doubt, keep the larger distance.

2.4 The base height

`base_height` levels the LiDAR's points (the height band of Chapter 5 counts from the floor) and places the near-field blind zone. Measure the height of base_link (the middle of the body, application manual 6.1) above the floor with the robot standing on level ground, to the centimetre.

CHAPTER 3

The driver: speeds to the robot

3.1 From a velocity to an axis command

`x30_driver` turns the mux's `cmd_vel` (base_link, REP-103: x forward, y left, counter-clockwise positive) into the robot's axis commands, streamed at 50 Hz to the motion host, only while the robot is in its stepping state and its motion report is fresh. Each axis is scaled to the maximum speed of the gait the robot walks in:

$$u = \text{round}\left(\frac{v}{v_{\max}(\text{gait})} \cdot 32767\right), \qquad |u| < 655 \;\Rightarrow\; u = \pm 656 \tag{3.1}$$

The lateral and yaw axes go out negated (the robot counts right as positive). A value inside the robot's dead zone ($|u| < 655$) would be ignored by the robot, so a small non-zero command goes out as the smallest value the robot moves on: the X30 creeps rather than stands.

Gait	Forward m/s	Backward m/s	Sideways m/s	Yaw rad/s	Slowest step forward m/s
Walk	1.2	0.75	0.15	0.45	0.024
Slope	0.7	0.75	0.25	0.5	0.014
Off-road	0.3	0.4	0.1	0.5	0.006
Stair	0.3	0.3	0.2	0.8	0.006
Stair (multi-frame)	0.6	0.3	0.2	0.8	0.012
Stair 45°	0.3	0.2	0.2	0.8	0.006
L-Walk	1.0	1.0	0.5	1.2	0.020
Mountain	1.0	1.0	0.8	1.2	0.020

Speeds at normal body height (application manual, Appendix B); at crawl height Walk and Slope go forward at half. Run and Silent have no published speeds: the driver does not move the robot in them unless `gait_limits` gives them, for example `gait_limits: {silent: [0.5, 0.5, 0.2, 0.5]}`.

> **The gait limits the stack** The robot profile's limits must hold in every gait the robot walks in during a patrol. In Off-road the X30 walks at most 0.3 m/s forward whatever the follower asks: the patrol is slower, still safe (the braking model assumed more), but the follower's progress checks wait longer. Keep Walk for patrols, or lower `cruise`.

3.2 The signs of the axes

A wrong sign makes the robot step right when the stack means left: every avoidance and the protective field's sideways check go wrong with it. Check once per robot, in REMOTE, with the robot standing in a free area and the handheld app closed (one source drives at a time):

```
rosservice call /control/set_mode "{mode: remote, source: handheld}"
rostopic pub -r 10 /control/remote/handheld/cmd_vel \
  geometry_msgs/Twist '{linear: {y: 0.1}}'
```

1. With the `rostopic pub` running, the robot must step to its **left**. If it steps right: `invert_vy: true` in `driver_params.yaml`.
2. The same with `'{angular: {z: 0.2}}'`: the robot must turn **counter-clockwise** (left). If not: `invert_wz: true`.
3. Ctrl-C, then back to AUTO: `rosservice call /control/set_mode "{mode: auto}"`.

3.3 Silence and stopping

`cmd_timeout_sec` (0.5 s) a command older than this is replaced by zeros;

`stop_burst_sec` (0.5 s) zeros streamed after motion ends, then silence (the robot itself drops an axis that falls silent for 1 s);

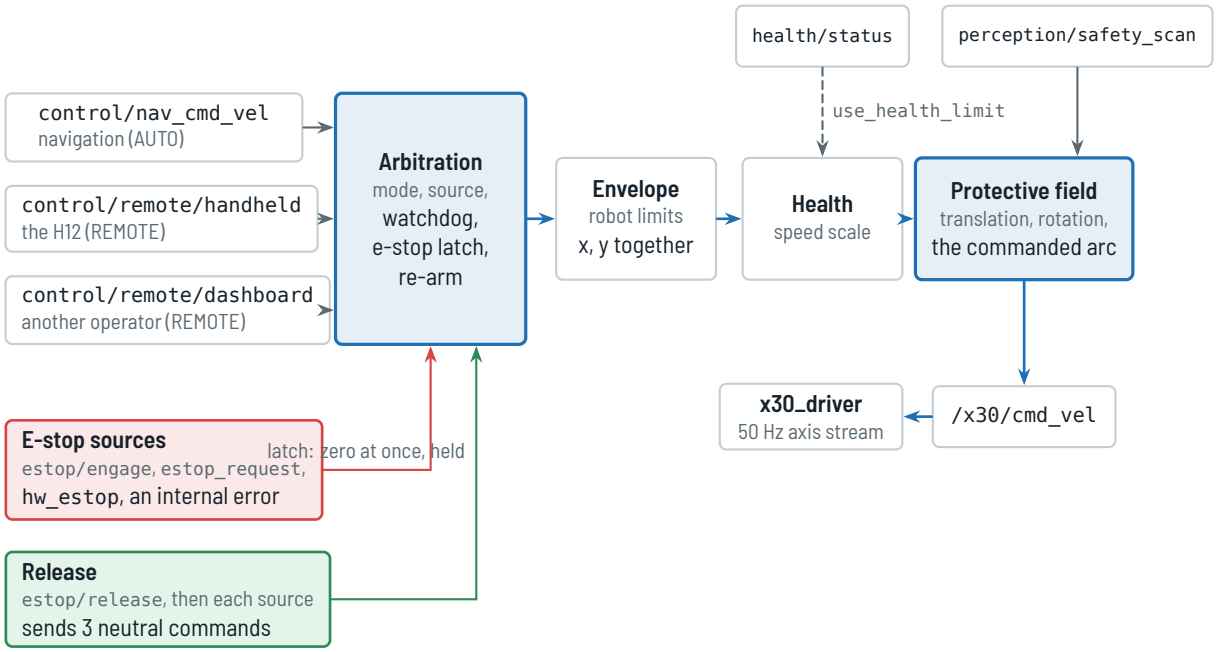
Episodes the first command after a quiet period starts an episode: the driver puts the robot into the control mode and the stepping state it needs (`auto_control_mode`, `auto_stepping`), three tries in two seconds. Nothing moves the robot after `sit_down`, an emergency stop, a fall or a takeover by Deep Robotics' handheld until the command has been zero or quiet once.

The driver's own clamps (`max_vx` 1.0 m/s, `max_vy` 0.5 m/s, `max_wz` 1.0 rad/s) are a last bound above the profile; the profile's limits are the ones the stack plans with.

CHAPTER 4

The control mux and the protective field

`control_mux_node` is the only publisher of the robot's `cmd_vel`. Every command, from navigation or from an operator, passes through it in this order:



4.1 Sources, silence and the e-stop

AUTO navigation drives (`control/nav_cmd_vel`); a silence longer than `nav_timeout_sec` (0.5 s) sends one zero.

REMOTE one operator source drives, each on `control/remote/<source>/cmd_vel`: `handheld` (the H12 gateway); `dashboard` is a second source nothing on the X30 publishes yet; a silence longer than `remote_timeout_sec` (0.3 s) sends one zero.

E-stop while engaged, zeros at `estop_hold_hz`. After the release, after a mode change or a source's silence, the source must first send `rearm_neutral_count` neutral commands in a row (every axis below `neutral_eps`): a stick held during the release never moves the robot.

4.2 The envelope

The command is brought inside the profile's limits, keeping its direction:

$$s = \min\left(1,\ \frac{v_{x,\max}}{|v_x|},\ \frac{v_{y,\max}}{|v_y|}\right), \qquad (v_x, v_y) \leftarrow s\,(v_x, v_y), \qquad \omega \leftarrow \mathrm{clip}(\omega, -\omega_{\max}, \omega_{\max}) \quad (4.1)$$

A non-finite value turns the whole command into zero. With `use_health_limit`, the health monitor's `speed_scale` (0 to 1) multiplies every command.

4.3 The protective field

The field limits each command to what the robot can still stop within, on the 360° safety scan (`perception/safety_scan`, Chapter 5). It applies in AUTO with `field.margin`, and in REMOTE with `field.remote_margin` unless `field.in_remote` is false.

4.3.1 Translation

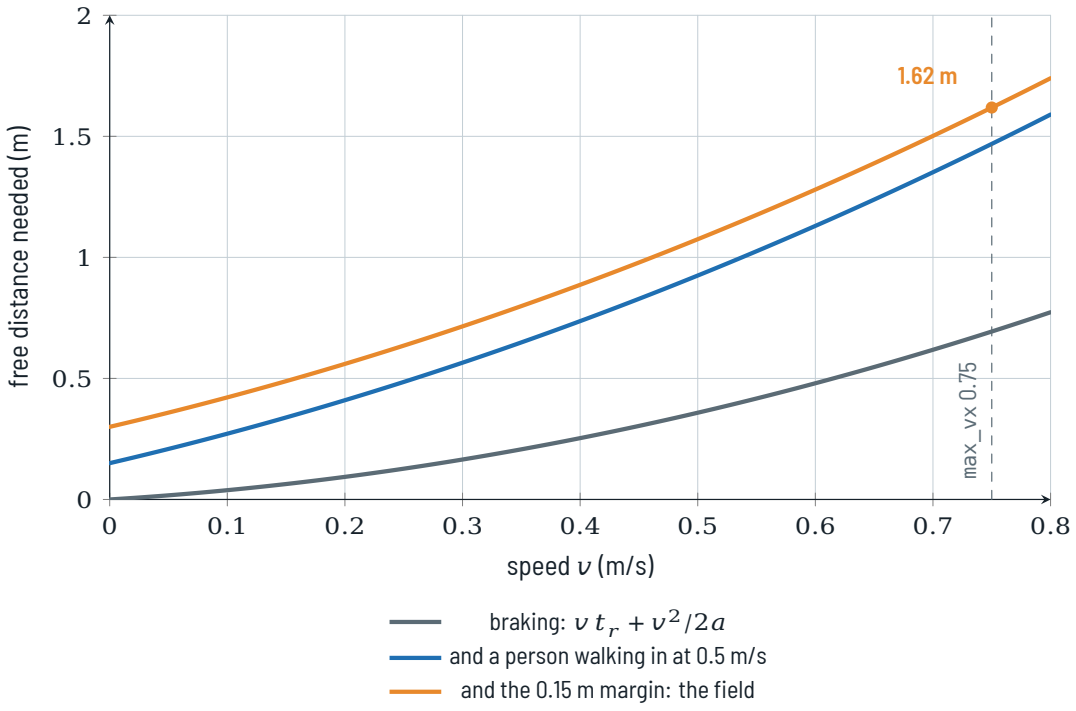
The footprint, grown sideways by `lateral_margin`, is swept along the commanded direction of travel; D is the free distance to the first obstacle point inside the swept band. The field keeps the speed and separation monitoring distance of ISO/TS 15066 inside D less the margin m: the robot's braking (Eq. 2.1) plus the distance a person walking toward the robot at v_h (`human_speed`) covers until the robot stands:

$$S(v) = v\,t_r + \frac{v^2}{2a} + v_h\left(t_r + \frac{v}{a}\right) \le D - m \tag{4.2}$$

Solved for the highest allowed speed:

$$v_{\text{allowed}} = a\left(-b + \sqrt{b^2 + \frac{2\,(D - m - v_h t_r)}{a}}\right), \qquad b = t_r + \frac{v_h}{a} \tag{4.3}$$

A faster command is scaled down to v_{allowed} in the same direction; an allowed speed below `min_speed` stops the translation instead (a crawl into an obstacle helps no one).



With the X30's profile (t_r 0.3 s, a 0.6 m/s^2, v_h 0.5 m/s):

Speed	Braking	Person	Field, AUTO	Field, REMOTE
m/s	m	m	m (m 0.15)	m (m 0.10)
0.10	0.04	0.23	0.42	0.37
0.25	0.13	0.36	0.64	0.59
0.50	0.36	0.57	1.08	1.03
0.75	0.69	0.78	1.62	1.57

Free ahead D	Allowed, AUTO
m	m/s
0.50	0.16
0.75	0.32
1.00	0.46
1.25	0.59
1.50	0.70
2.00	0.90

4.3.2 Rotation

For a turn, the field finds the angle θ the footprint can rotate before it touches a point, less `yaw_margin`, and allows the yaw rate that stops within it:

$$\omega_{\text{allowed}} = \alpha \left(-t_r + \sqrt{t_r^2 + \frac{2\,\theta}{\alpha}} \right) \tag{4.4}$$

Below `min_yaw_rate` the turn stops. A command that translates and turns at once is checked along its stopping arc: the footprint at every step of the arc until both axes stand, with the margin grown by the person's approach during the braking time $v_h\ (t_r + \max(v/a, |\omega|/\alpha))$, and the whole command scaled down until the arc stays clear.

4.3.3 When the scan cannot be trusted

State	Behaviour
STALE	no scan newer than `scan_timeout_sec`, or its stamp older than `scan_max_age_sec`: AUTO stops; REMOTE creeps, the fastest point of the body at `remote_degraded_speed`
BLIND	the LiDAR sees too little (Chapter 5): AUTO stops; REMOTE creeps and still keeps clear of what is seen, never toward a blind sector

4.4 Tuning the field

`margin` what the model does not know: the scan's motion smear and its sector discretization (Chapter 5), the robot's sway while walking. 0.15 m covers the X30 at 0.75 m/s on a 10 Hz scan; raise it on rough ground.

`human_speed` the trade-off of the whole field. 0.5 m/s presumes trained staff who see the robot. ISO 13855 assumes 1.6 m/s for a walking person: the field at 0.75 m/s then needs 3.3 m free ahead, and the robot slows wherever people may walk. Choose it with the site's risk assessment, never lower than the people near the robot actually walk.

`lateral_margin` the sideways allowance of the swept band; the follower's guard must use at least the same (`path_follower_node/safety/lateral_margin`).

`min_speed`, `min_yaw_rate` below the X30's slowest steps (Chapter 3) a command does not move the robot anyway.

> **The field is software** The protective field is not a certified safety function. It reduces risk; the hard emergency stop and a responsible operator stay the protection.

CHAPTER 5

Perception: the sector scans

`lidar_sector_node` (navigation host, on Deep Robotics' own LiDAR driver there) turns every LiDAR cloud into two sector scans: the range of the nearest obstacle in each angular sector around the robot.

Scan	Sectors	Read by
`perception/safety_scan`	72 over 360°, 0.1 m to 4 m	the protective field and the follower's guard
`perception/sector_scan`	32 over 180° ahead, 0.1 m to 8 m	reactive avoidance

5.1 Which points are obstacles

Each cloud goes through four filters before it is binned:

1. **Noise**: Livox tags every point with its confidence. Adhesion points (bits 0–1, the glue of two surfaces) and particle points (bits 2–3: rain, dust) at the confidence of `noise_level` are dropped.
2. **Self**: points inside the footprint grown by `self_margin` are the robot.
3. **Levelling**: the cloud is rotated by the base's roll and pitch (from odom, TF), so heights count from the floor even while the body pitches in its gait. Without a recent roll and pitch (`tilt_max_age`) every sector is BLOCKED: an unlevelled cloud could put the floor in the band.
4. **Height band**: only points between `height_min` (0.10 m) and `height_max` (1.60 m) above the floor are obstacles. With `ground_segmentation` the floor height is estimated per `ground_cell` first, for ramps and uneven yards.

5.1.1 Binning

A sector's range is its k-th nearest point (k = 2): one stray return never blocks a sector, and a sector needs k points to count. Beyond `range_max` a sector is free; a non-finite or blocked range is treated by every consumer as an obstacle at the robot.

5.2 A blind LiDAR

A covered window, a failed sensor or heavy rain empties the cloud. Fewer than `min_points` (100) valid points in a cloud, or with `regions` and `min_region_points` fewer than the minimum in any azimuth region, make the cloud blind: every sector BLOCKED, the field in its BLIND state (Section 4.3). No cloud for `timeout_sec` does the same.

> **A partly covered sensor** `min_region_points` catches what the total count misses: a bag on one side of the robot leaves enough points overall. To find the normal count, set `min_region_points` very high for a moment at the site: the warning `scans BLOCKED ... N points in region k` names the weakest region's count. Then set the minimum at a third of it (a start-only parameter: restart).

5.3 The near-field blind zone

A LiDAR sees nothing below its lowest beam. With the sensor at height H above the floor and its steepest beam δ below the horizontal, an object of height h disappears from the cloud once it is closer than

$$r(h) = \frac{H - h}{\tan \delta} \tag{5.1}$$

The robot walking toward a low obstacle sees it from afar, then loses it right where it matters. The **obstacle memory** keeps every obstacle cell in the odom frame for `memory_sec` (10 s) and adds it back to the scans while it is below the view, within `radius`. The model takes H from `base_height` and the LiDAR's mount (`lidar_pose`), and δ from `lower_fov_deg` and the mount's tilt.

	blind radius $r(h)$ for an object of height		
Lowest beam δ	0.10 m	0.20 m	0.30 m
7° (Mid-360, level)	3.1 m	2.3 m	1.5 m
30°	0.66 m	0.48 m	0.31 m

($H = 0.48$ m, the sensor at base_link height.) The X30's downward LiDARs see much closer than a level sensor: set `lower_fov_deg` to the steepest beam of the cloud the node reads, and check with a box of the height of `height_min` that the scan keeps it while the robot walks up to it.

> **The memory forgets on a jump** An odometry jump larger than the motion (`max_jump`) clears the memory: remembered cells would land in the wrong place. A restarted odometry does the same.

5.4 Motion smear

The Livox accumulates a cloud over $T = 0.1$ s while the robot moves. The cloud is read as it arrives, so a point measured at the start of the sweep is off by up to

$$\Delta \approx v\,T + \omega\,T\,r \tag{5.2}$$

for a point at range r: 7.5 cm straight ahead at 0.75 m/s, 14 cm at 2 m while walking at 0.5 m/s and turning at 0.45 rad/s. The protective field's `margin` and the guard's `lateral_margin` must cover it. To remove it instead, read FAST-LIO's undistorted cloud (`input.type: pointcloud2`, `input.pointcloud_topic: /cloud_registered_body`) with its frame in TF; it arrives one LIO step later.

5.5 Tuning the band

`height_min` above the floor's unevenness and the gravel, below the lowest thing that can trip the robot. Points of a slope rise into the band: enable `ground_segmentation` on ramps rather than raise the bottom.

`height_max` above the highest part of the robot with its payload (the PTZ camera on its mast): anything lower that hangs over the route must be seen.

`range_max` (safety) at least the field's distance at full speed plus the footprint: 1.62 m + 0.50 m for the X30; 4 m leaves room.

`num_sectors` 72 gives 5° sectors: at 2 m a sector is 17 cm wide. More sectors resolve a narrow gap better and need more points each.

CHAPTER 6

Odometry

FAST-LIO, a LiDAR-inertial odometry, gives the robot's motion (`/Odometry`); the X30's leg odometry from the motion host is the second opinion. `x30_odometry` holds FAST-LIO back until the robot stands still, and watches it while it runs.

6.1 The start gate

FAST-LIO takes gravity and the IMU's biases from its first second of data. Started while the robot walks, sways or stands on a slope, its frame is tilted and every height of the stack is wrong with it. `wait_still` starts FAST-LIO only once the IMU has been still for `still_sec` (2 s):

Condition	Default	Meaning
`max_gyro_rad_s`	0.03 rad/s	no turning
`max_acc_rel_std`	0.01	$\sigma(\lvert a\rvert)/\lvert\overline{a}\rvert$: no shaking (unit-free)
`max_tilt_deg`	3°	gravity within 3° of the IMU's z axis: standing level
`min_samples`	20	IMU samples in the window
`timeout_sec`	60 s	then start anyway, with an error in the log

Start on the dock The robot standing on its dock, sitting or standing still, passes the gate in two seconds. On a slope above 3° the gate waits for its timeout: start the robot on level ground.

6.2 The monitor

`odometry_monitor_node` publishes `odometry/status`: OK, DEGRADED or LOST, with the reason. The health monitor makes LOST an error and DEGRADED a warning.

State	When
LOST	no pose for `stale_sec` (0.5 s); a jump or a restart, for `hold_sec` (5 s) after it
DEGRADED	the speed residual alarms; the odom frame tilted against gravity by more than `tilt.max_deg` (measured while still); $\lvert z\rvert$ above `max_z_drift_m` on flat ground
OK	none of these

6.2.1 Jumps and restarts

A step between two poses is a **jump** when it is longer than `jump.min_step_m` (0.3 m) and faster than `jump.speed_factor` times the robot's speed limit: no gait covers it. A **restart** is a message gap of `restart.gap_sec` followed by a pose at the origin (within `origin_radius_m` and `origin_yaw_deg`): FAST-LIO began again, and every pose in odom before it is meaningless (Chapter 7).

6.2.2 The speed residual

FAST-LIO's speed, from its poses over `window_sec`, is compared with the leg odometry's planar speed. The residual, normalized by both uncertainties, is a χ^2 sample with one degree of freedom:

$$r = v_{\text{LIO}} - v_{\text{leg}}, \qquad \chi^2 = \frac{r^2}{\sigma_{\text{LIO}}^2 + \sigma_{\text{leg}}^2} \tag{6.1}$$

One sample means little (the legs slip, the gait sways), a persistent disagreement means a lot. A one-sided CUSUM accumulates the excess over κ and alarms above h:

$$s_k = \min\!\big(2h,\ \max(0,\ s_{k-1} + \chi_k^2 - \kappa)\big), \qquad \text{alarm while } s_k > h \tag{6.2}$$

With $\kappa = 3.84$ (the 95 % point of χ_1^2), $h = 20$, $\sigma_{\text{LIO}} = 0.05$ m/s and $\sigma_{\text{leg}} = 0.15$ m/s:

- a residual below $\sqrt{\kappa\,(\sigma_{\text{LIO}}^2 + \sigma_{\text{leg}}^2)} = 0.31$ m/s never alarms, however long it lasts;
- a residual of 0.5 m/s adds $10.0 - 3.84 = 6.2$ per pose and alarms at the fourth;
- the cap $2h$ bounds the recovery: after the disagreement ends, the alarm clears within a few poses.

Raise σ_{leg} on gravel, mud or wet grass, where the legs slip: false alarms vanish, real ones come later.

Lower h to catch a degenerate FAST-LIO (a featureless yard, a long uniform wall) sooner, at the cost of more alarms on rough ground.

6.3 The trip meter

`odometry/trip` counts the distance, the moving time and the speeds from FAST-LIO's poses since the last reset (`odometry/trip_reset`, the app's long press). Motion is counted in steps of `step_m`, so the sway of a standing robot adds nothing; a speed above `max_mps` is a jump, not a trip.

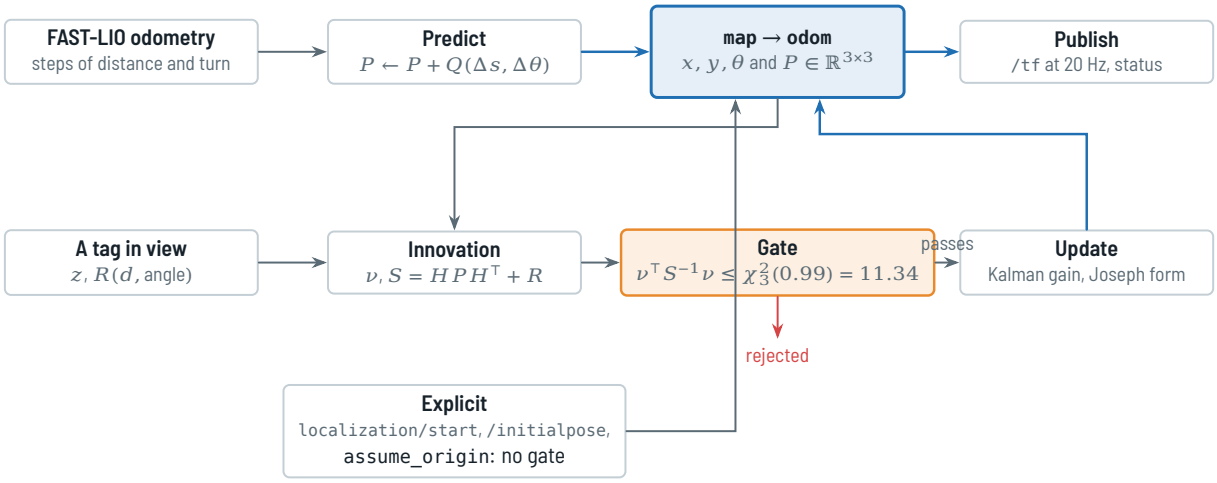
CHAPTER 7

Localization

7.1 No map: the start spot and the tags

The X30 patrols without a map. The robot starts on its dock; FAST-LIO's start is the origin of the `map` frame, and every route is taught and repeated in that frame. Odometry drifts with every metre; **tags** (printed ArUco markers on walls and posts, recorded once in the site) correct the drift wherever the robot sees one.

`localization_node` estimates ${}^{map}\mathbf{T}_{odom}$, the correction between the map and the odometry frame, with an extended Kalman filter on $SE(2)$:



7.2 The filter

7.2.1 State and prediction

The state is $x = (x, y, \psi)$ of ${}^{map}\mathbf{T}_{odom}$ with its covariance $\mathbf{P}$. Between corrections, odometry carries the robot and x stays; only the uncertainty grows, and only with motion. Motion is counted in steps (`motion_step_m` 5 cm, `motion_step_deg` 2°) so that the sway of a standing robot adds nothing. For a step of Δs metres and $\Delta\theta$ radians, the process noise in the robot frame is

$$\mathbf{Q}_{\text{base}} = \operatorname{diag}\left(\sigma_{xy}^2\,\Delta s,\ \sigma_{xy}^2\,\Delta s,\ \sigma_{\psi,m}^2\,\Delta s + \sigma_{\psi,\text{rad}}^2\,|\Delta\theta|\right) \tag{7.1}$$

mapped into the state through the inverse of the measurement Jacobian, $\mathbf{P} \leftarrow \mathbf{P} + \mathbf{H}^{-1}\mathbf{Q}_{\text{base}}\mathbf{H}^{-\top}$. With the defaults ($\sigma_{xy} = 0.03\,\text{m}/\sqrt{\text{m}}$, $\sigma_{\psi,m} = 0.1°/\sqrt{\text{m}}$) a 50 m walk without a tag leaves $\sigma_{xy} = 0.03\sqrt{50} = 0.21$ m and $\sigma_{\psi} = 0.71°$.

7.2.2 Correction

A tag seen from the robot gives a measurement z of ${}^{map}\mathbf{T}_{base}$ with covariance $\mathbf{R}$. The innovation against the prediction $h(x) = {}^{map}\mathbf{T}_{odom}\,{}^{odom}\mathbf{T}_{base}$, its covariance and the squared Mahalanobis distance are

$$\nu = z \ominus h(x), \qquad \mathbf{S} = \mathbf{H}\mathbf{P}\mathbf{H}^{\top} + \mathbf{R}, \qquad d^2 = \nu^{\top}\mathbf{S}^{-1}\nu \tag{7.2}$$

A measurement with d^2 above the χ^2_3 gate is rejected as an outlier (a tag recorded wrong, a tag moved, a false detection). Otherwise

$$\mathbf{K} = \mathbf{P}\mathbf{H}^\top \mathbf{S}^{-1}, \qquad x \leftarrow x + \mathbf{K}\nu, \qquad \mathbf{P} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\,\mathbf{P}\,(\mathbf{I} - \mathbf{K}\mathbf{H})^\top + \mathbf{K}\mathbf{R}\mathbf{K}^\top \tag{7.3}$$

The Joseph form keeps **P** symmetric and positive definite. After the update the uncertainty is floored at `min_sigma_xy` (1 cm) and `min_sigma_yaw_deg` (0.25°): the filter never becomes so sure of itself that it rejects the next tag.

`gate_confidence`	0.95	**0.99**	0.999
χ^2_3 gate	7.81	**11.34**	16.27

7.2.3 How much a tag is trusted

A tag's pose is less certain the farther and the more obliquely it is seen. For a tag at distance d seen at angle θ off its normal:

$$\sigma_p = \sigma_{p,1}\,\frac{d^2}{\cos\theta}, \qquad \sigma_\psi = \sigma_{\psi,1}\,\frac{d}{\cos\theta} \tag{7.4}$$

with $\sigma_{p,1}$ 2 cm and $\sigma_{\psi,1}$ 1° at one metre (`sigma_xy_at_1m`, `sigma_yaw_deg_at_1m`). A heading error also moves the robot about the tag, so **R** couples position and heading through the lever arm to the tag. Tags farther than `max_distance_m` (4 m) or seen beyond `max_view_angle_deg` (60°) are ignored: at 4 m a tag's position is already worth 32 cm.

7.3 The states

NONE no ${}^{map}\mathbf{T}_{odom}$ yet: navigation does not start.

LOCALIZED the robot knows where it is: on the start spot (`origin_sigma_xy` 5 cm, `origin_sigma_yaw_deg` 2°), or from a tag.

UNCERTAIN σ_{xy} above `max_sigma_xy` or σ_ψ above `max_sigma_yaw_deg`.

The follower pauses only at NONE. UNCERTAIN is reported on `localization/status` and by the health monitor, and the patrol goes on: without a map the robot has nothing better than its odometry between tags. The defaults (100 m, 180°) never leave LOCALIZED; lower them to make a long walk without tags visible in NETRA and the health report.

7.4 Tags

Recording `localization/record_tags` averages each visible tag over `record_duration_sec` (2 s) and keeps it only with `record_min_samples` detections within `record_max_std_m` and `record_max_std_deg`. Record from where the robot will see the tag on its routes, 1 m to 2.5 m away, as square-on as possible.

Automatic `auto_localize`: while not localized, any visible recorded tag localizes the robot. `auto_correct`: while localized, visible tags are fused through the gate.

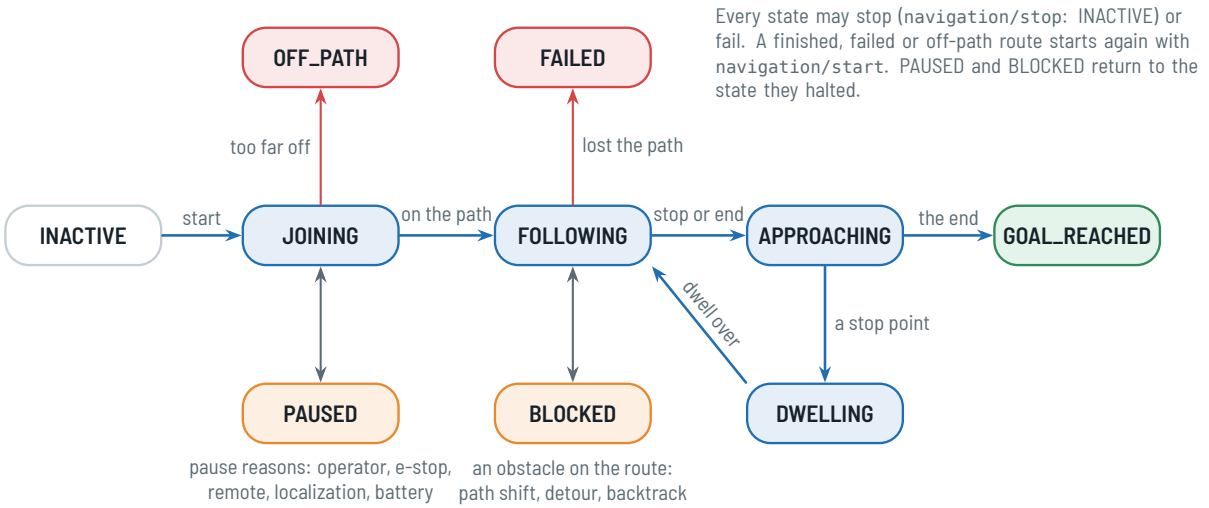
Placement one tag every 20 m to 40 m of route, at the camera's height, where the robot looks at it while walking (the start of a straight, a stop point), never on something that moves (a gate, a door).

A restarted odometry FAST-LIO restarting (Chapter 6) resets the odom frame: the old ${}^{map}\mathbf{T}_{odom}$ means nothing. The localization drops to NONE and the patrol pauses. Show the robot a recorded tag (`localization/start`, the app's **Localize**), or bring it back to its start spot and restart the navigation host's service: the start spot becomes the origin again. (`localization/assume_origin` takes the current odom frame as the map: right only when FAST-LIO restarted on the start spot.)

CHAPTER 8

Following a route

`path_follower_node` walks the robot along a taught route, stops and dwells at its stop points, and reports its state on `navigation/status`.



8.1 From a recording to a route

Teaching records the localized pose every `min_spacing_m` (0.2 m). Loading the route then cleans it: duplicate points go, the route is resampled every `processing.spacing` (0.1 m) and smoothed between anchors that never move (the ends and the stop points). The smoothed points y_i are the fixed point of the gradient smoother on the recording x_i:

$$\alpha\,(x_i - y_i) + \beta\,(y_{i-1} - 2y_i + y_{i+1}) = 0, \qquad \alpha = w_d\,(1 - 2w_s), \quad \beta = w_s \tag{8.1}$$

solved at once as a tridiagonal system, with w_d `smooth_weight_data` (0.05) and w_s `smooth_weight_smooth` (0.4). A larger w_s cuts the operator's wobbles and the corners more. The node warns when a point moves more than `max_deviation` (0.3 m): the robot would leave the walkway that was taught.

8.2 The speed profile

Each point of the route gets a speed limit, computed once when the route loads:

1. **Curves**: with κ the curvature, $v \le \sqrt{a_{\text{lat}}/\kappa}$, clipped to [`min_speed`, `cruise`] (a_{lat}: `lateral_accel`).
2. **Stops**: stop points and the end at `approach_speed`.
3. **Braking into them**, backwards along the route: $v_i \le \sqrt{v_{i+1}^2 + 2\,a_{\text{dec}}\,\Delta s_i}$, and the reaction time of the braking model (Eq. 2.2) before each stop.
4. **Accelerating out**, forwards: $v_{i+1} \le \sqrt{v_i^2 + 2\,a_{\text{acc}}\,\Delta s_i}$.

Curve radius	0.5 m	1 m	2 m	straight
Limit (a_{lat} 0.3 m/s^2)	0.39 m/s	0.55 m/s	0.77 m/s	`cruise`

The X30's `cruise` is 0.4 m/s: only curves tighter than about 0.5 m slow it. `accel` and `decel` (0.3 m/s^2) are comfort values, below the profile's braking: the robot slows into a stop point gently, not at the field's limit.

8.3 The path controllers

Every controller gets the robot's pose, the route, its progress s along it, the speed of the profile and the path shift offset, and returns (v_x, v_y, ω). `controller` picks one; the X30 uses `holonomic`.

8.3.1 Holonomic (the X30's)

A carrot is taken `lookahead` (0.6 m) ahead along the route. The robot walks toward it in its own frame at $\min(v, d_{\text{carrot}})$, and turns toward the route's direction over the next `heading_lookahead`:

$$\omega = k_\psi\, e_\psi, \qquad (v_x, v_y) = \min(v, d)\, \hat{u}_{\text{carrot}} \cdot \max(\texttt{min_alignment_scale},\ \cos e_\psi) \tag{8.2}$$

with k_ψ `kp_yaw` (1.5 1/s). The sideways part is capped at `max_lateral_speed` (above the X30's Walk gait the envelope caps it again at `max_vy`), the backward part at `max_reverse_speed`. A longer lookahead cuts corners and sways less; a shorter one tracks tighter.

8.3.2 The others

`pure_pursuit` the arc through the carrot at lookahead L: with the carrot at (x, y) in the robot frame, $\kappa = 2y/(x^2 + y^2)$ and $\omega = v\,\kappa$.

`stanley` the route's curvature as feedforward plus the cross-track error e and the heading error e_ψ: $\omega = v\,\kappa_{\text{route}} - k_h\,(e_\psi + \arctan\frac{k\,e}{v + k_s})$. k_s keeps low speeds from over-steering.

`pid` PID on the heading error toward the carrot, the integral clamped at `integral_limit` and frozen while the output saturates (anti-windup).

`lqr` the path error $e = (e, e_\psi)$ of a unicycle at speed v, discretized at the control period T: $e_{k+1} = \left(\begin{smallmatrix}1 & vT\\ 0 & 1\end{smallmatrix}\right) e_k + \left(\begin{smallmatrix}0\\ T\end{smallmatrix}\right)\omega_k$. The gain $\mathbf{K}$ solves the discrete Riccati equation for $\mathbf{Q} = \text{diag}(q_e, q_\psi)$ and r_ω (cached every 5 cm/s of speed); $\omega = v\,\kappa_{\text{route}} - \mathbf{K}e$.

`mppi` `samples` yaw-rate sequences drawn around the previous plan, each rolled out as a unicycle over `horizon_steps`; the cost is the distance to the route, the distance of the end to the carrot and `obstacle_weight` times the deepest intrusion into `clearance`. The sequences are averaged with the weights $w_j \propto \exp(-(J_j - J_{\min})/\lambda)$, λ `temperature`.

`neupan` NeuPAN plans forward, sideways and yaw motion over a horizon straight from the scan's points, with a network (DUNE) trained for the X30's footprint. It avoids by itself; the guard and the field stay. It needs its own Python environment (`tools/install_neupan.sh`).

8.4 Stops, joining and dwelling

Joining a robot up to `join.max_distance` (3 m) from the route first walks onto it, within `join.on_route`.

Approach `approach.radius` before a stop point the follower leaves the controller and converges onto the exact pose: $v = k_{xy}\, e_{xy}$, $\omega = k_\psi\, e_\psi$, capped at `approach.max_speed` and `approach.max_yaw_rate` and never below `approach.min_speed` (the X30's slowest step). It arrives within `position_tolerance` (5 cm) and `yaw_tolerance_deg` (3°); a stop the robot passed by more than `skip_margin` counts as skipped.

Dwelling at the stop the inspection runs (`hooks.inspection`); the follower waits for it up to the dwell plus `hooks.inspection_timeout_sec` (180 s on the X30, above the inspection's own 150 s per point).

Battery below `low_battery` the lap finishes and no new one starts; below `critical_battery` the robot pauses.

8.5 Smoothness

The command passes a last stage before it leaves: the profile's envelope, a smoothing of the yaw rate and the sideways speed, $\omega \leftarrow w\,\omega_{\text{new}} + (1 - w)\,\omega$ with w `speed.smoothing` (0.5; 1: off), and the acceleration limits. A robot that weaves on a straight needs a lower w or a longer lookahead; one that lags in curves, the opposite.

8.6 The supervisor

Setting	Effect
`tf_timeout_sec`, `scan_timeout_sec`	an older pose or safety scan stops the robot
`off_path_slow`, `off_path_stop`	farther from the route than 0.6 m: the floor speed; than 1.5 m: OFF_PATH, stopped
`blocked_timeout_sec`	standing still (moved less than `stall_distance` and turned less than `stall_yaw_deg`) for 10 s: BLOCKED, reported to NETRA
`lateral_margin`, `human_speed`, `pass_margin`, `yaw_margin_deg`	the guard's model; at least the protective field's values, checked at start against `control_node`

CHAPTER 9

Getting past obstacles

A substation's walkways are narrow and lined with energized equipment: getting past a bucket or a person must never mean walking off the walkway. The X30 has five layers, from the gentlest to the last resort:

Layer	What it does
Path shift	slides sideways off the route and back, keeping the heading (holonomic robots)
Detour	stuck: turns toward a gap and walks a short straight through it
Backing up	stuck for long: backs up along the driven path and tries again
The guard	the follower stops its own command before it could hit anything
The field	the control mux stops whatever is left (Chapter 4)

9.1 Path shift

The free corridor along the route ahead (`look_ahead`, 2 m beyond the front edge) is evaluated in route coordinates for candidate lateral offsets every `step` (0.1 m) up to `max_offset` (0.8 m) on either side, the body plus `clearance` beside it. The robot takes the free offset nearest the one it holds: a new side must be clearer by `switch_margin` and pays `switch_cost` (hysteresis), so it does not hesitate between left and right. It slides onto the offset at `offset_rate`, and comes back onto the route once the way back has stayed free for `return_delay_sec`.

When no offset is free, the robot slows to the speed that stops `stop_margin` before the obstacle (Eq. 2.2 on the free distance less the margin) and waits. After `patience_sec` it searches wider, up to `search_offset` (1.5 m). A lane with less than `centre_room` beside the body is walked in its middle.

> **Set the offsets to the walkway** `max_offset` and `search_offset` are how far the robot may leave the taught route. Next to energized equipment they must be smaller than the room the walkway gives on either side, less the minimum approach distance. A walkway 1.5 m wide taught along its middle leaves $(1.5 - 0.47)/2 \approx 0.5$ m: set `max_offset` and `search_offset` to 0.5 m or less there, or teach the route where there is room.

9.2 Detours and backing up

Detour stuck for `detour.after_sec` (1.5 s) while following, the robot turns toward a gap within `max_angle_deg` (75°) of the route's direction and walks straight through it, at most `length` (2 m) at `speed`, only if the run brings it `min_progress` along the route. A detour leaves the route more than path shift does: disable it (`detour.enabled: false`) where the walkway leaves no room.

Backing up no progress along the route for `backtrack.after_sec` (30 s): the robot backs up `distance` (1 m) along the path it walked, with the guard on the rear sectors, and tries again; after `max_attempts` at one place it stays BLOCKED and NETRA gets the alert.

9.3 Reactive avoidance

`obstacle_avoidance_node` reads the front sector scan and turns it into a bend of the command: a heading offset, a sideways bias and a speed scale. With `avoidance_mode: reactive` the follower applies them; in every mode but `off`, its **emergency** stops the forward motion: an obstacle within `emergency_distance` (0.75 m) in the frontal `emergency_cone_deg` (60°). With path shift and `emergency_slide`, the robot may still slide out sideways.

Every policy works on the danger of each sector, from 0 beyond `safe_distance` r_s (1.5 m) to 1 at the robot:

$$\delta_j = \mathrm{clip}\left(1 - \frac{r_j}{r_s},\ 0,\ 1\right) \tag{9.1}$$

`vfh` (the X30's) the headings whose sectors are free, in runs at least `min_gap_width_deg` wide; it takes the free heading nearest straight ahead and slows by the frontal danger: $s = \max(0, 1 - k_v\ \delta_{\text{front}})$.

`follow_gap` removes a bubble of `bubble_radius` around the nearest obstacle (half-angle $\arctan(r_b / r_{\min})$) and aims at the centre of the widest remaining gap.

`potential_field` each obstacle inside r_s pushes with $F = k_r\ (1/r - 1/r_s)/r^2$; the sideways sum steers (k_θ arctan F_y) and pushes (`lateral_gain`, capped at `max_lateral`).

`braitenberg` the danger left and right (triangular weights around `steering_zone_center_deg`) steers away: $\Delta\theta = k_s(\delta_R - \delta_L)$; the danger ahead and beside slows: $s = \max(0, 1 - k_v \delta_{\text{side}})(1 - \delta_{\text{ahead}})$.

The outputs are smoothed with `shared.smoothing` toward steering and speeding up, never toward braking: a stop is never delayed.

9.4 The guard

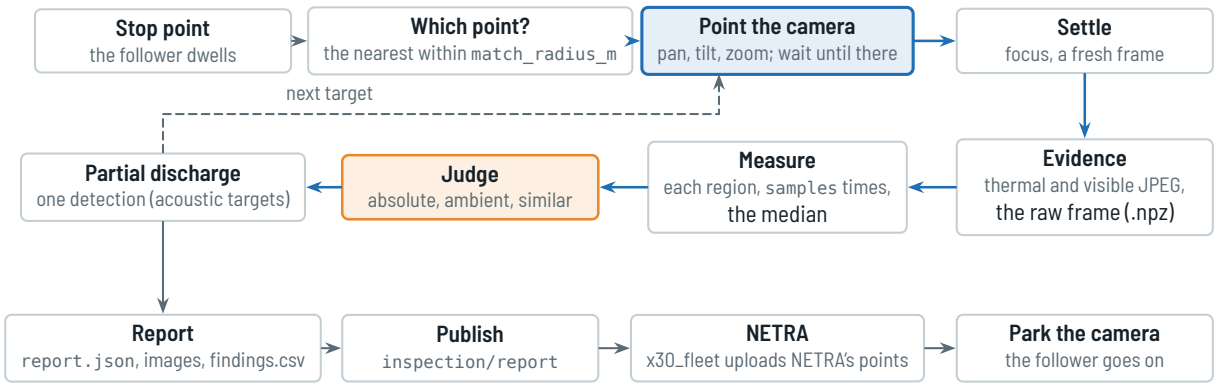
Before it publishes, the follower checks its own command against the safety scan with the same braking and separation model as the field, with its own margins (`safety.*`, Chapter 8). The guard's margins are at least the field's, and the follower checks that at start: the follower stops itself before the field ever has to. A field that limits the follower's commands in normal operation means the two are tuned apart.

CHAPTER 10

Inspection and recording

10.1 At a stop point

`inspection_node` runs when the follower reaches a stop point: it finds the inspection points of the site within `match_radius_m` (1.5 m) of the stop, and for each target moves the PTZ camera, waits, measures and judges.



Setting	Default	Tune when
`camera.move_timeout_sec`	20 s	a long pan fails: the camera is slow in the cold
`camera.settle_sec`	1 s	images blur or the focus hunts: raise to 2 s to 3 s at high zoom
`thermal.samples`	3	readings jump: more samples (the median is used)
`thermal.sample_interval_sec`	0.3 s	the frames are the same: longer
`max_duration_sec`	150 s	a point has many targets: later targets are skipped
`acoustic.duration_sec`	5 s	the acoustic camera needs longer to classify

Note `max_duration_sec` must stay below the follower's `hooks.inspection_timeout_sec` (180 s on the X30): otherwise the robot walks on while the camera still works.

10.2 Thermal criteria

Each target has one or more regions on the thermal image. With $T_{\max}$ the hottest region and T_a the air temperature (`thermal.ambient_c`), three differences are judged against their levels, after NETA MTS practice:

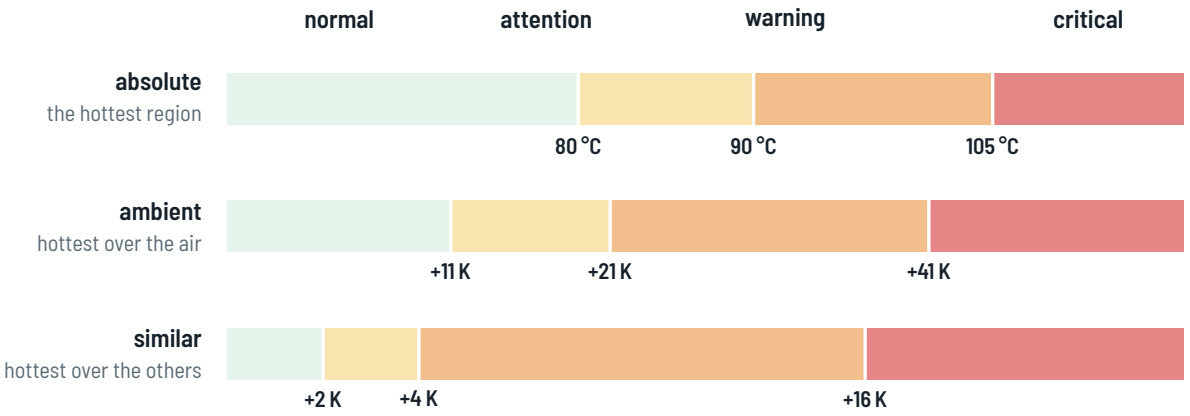
$$\text{absolute:} \quad T_{\max} \tag{10.1}$$

$$\text{ambient:} \quad \Delta T_a = T_{\max} - T_a \tag{10.2}$$

$$\text{similar:} \quad \Delta T_s = T_{\max} - \text{median}\{T_j : j \neq \max\} \qquad \text{(two regions or more)} \tag{10.3}$$

Criterion	Attention	Warning	Critical
absolute $T_{\max}$	80 °C	90 °C	105 °C (or the target's `max_temp_c`)
ambient ΔT_a	11 K	21 K	41 K
similar ΔT_s	2 K	4 K	16 K

The target's severity is the highest of its three. A target taught with a rated temperature (`max_temp_c` in NETRA) uses it as its absolute critical level.



A finding takes the highest severity of its criteria; a target's own `max_temp_c` replaces 105 °C as its critical limit. Scales are not to scale between the rows.

The air temperature is live: set it ΔT_a is only as good as T_a. The default 30 °C misjudges a hot afternoon or a cool night by the difference. Set it each day (it takes effect within a second): `rosparam set /inspection_node/thermal/ambient_c 34`.

10.2.1 Getting good temperatures

- **Emissivity**: the camera's setting applies to every target. Bare, oxidized copper and aluminium read far too cold; the similar criterion compares like with like and suffers least.
- **Distance and zoom**: a region needs several pixels of the hot spot; zoom in from the stop rather than teach a closer stop next to the equipment.
- **Sun and reflections**: a reflection of the sun on a busbar is hot to the camera. Patrol thermal routes at night or on a cloudy day where it matters.
- **Load**: a joint's heating rises with the square of the current. Readings at light load understate a defect; NETRA's trend over patrols shows it best.

10.3 Partial discharge

The acoustic camera classifies what it hears; each class maps to a severity: `acoustic.corona` (attention), `acoustic.surface` (warning), `acoustic.floating` (critical), noise is normal. `intensity_warning` and `intensity_critical` raise a discharge by its intensity as well (0: off): set them after a few patrols, from the intensities the site's normal corona shows.

10.4 Reports

Every point inspected writes a folder under `reports_dir` with the readings, the images and, with `camera.raw_thermal`, the raw temperatures per pixel. The oldest days go while less than `reports_min_free_gb` (6 GB) is free; `reports_keep_days` bounds them by age.

10.5 The recorder

`recorder_node` (perception host) records the cameras during every patrol (`patrol.enabled`) and on request: the visible and thermal streams as MP4 segments of `video.segment_sec`, every raw thermal frame with the robot's pose and the camera's pan, tilt and zoom, and the acoustic data in chunks.

Setting	Default	Meaning
`disk.min_free_gb`	10 GB	the oldest sessions go while less is free
`disk.stop_free_gb`	3 GB	below this nothing more is written
`disk.check_sec`	30 s	checked this often while recording
`patrol.stop_after_sec`	30 s	after the route ends; laps stay one session
`thermal.min_interval_sec`	0	every thermal frame; raise it to save disk

A raw thermal frame holds a temperature for every pixel: at the camera's full frame rate an hour of patrol fills gigabytes. Set `thermal.min_interval_sec` to 1 s or more when the raw frames serve trends rather than video.

CHAPTER 11

Health, the handheld and NETRA

11.1 The health monitor

`health_monitor_node` publishes `health/status` every second (and `/diagnostics`): every finding with its level, and a speed scale.

Check	Issue when
`nodes.expected`	a node of the host is missing (filled by bringup from what it starts)
`sensors.topics`	a topic below `rate_tolerance` of its rate (`/livox/lidar` 5 Hz, `/Odometry` 8 Hz) or its stamps too old
`state.*`	odometry LOST (error) or DEGRADED, localization NONE, the e-stop engaged or the protective field stale or blind (errors)
`tf.*`	a dynamic transform older than `max_age_sec`, or from the future (clocks)
`system.*`	the hottest thermal zone above 85 °C, memory above 92 %, less than 5 GB free, a load above 2 per CPU
`clock.*`	(off) `chronyc tracking` offset above 50 ms

For `startup_grace_sec` (30 s) after the start, missing nodes and inputs only warn.

11.1.1 The speed scale

An error makes the scale 0; a warning takes the lowest scale of the `degradation.rules` it matches (`sensor:/livox/lidar=0.5`, `sensor:/Odometry=0.5`, `tf:*=0.5`), otherwise `warn_scale` (1). The control mux applies it to every command only with `use_health_limit: true` in `x30_control/config/control.yaml`; the X30 ships with it off, and the follower pauses by itself on the conditions that matter (stale poses and scans, NONE). Turn it on to make a slow LiDAR or odometry halve the speed instead of merely reporting it.

11.2 The handheld gateway

`teleop_udp_node` serves the H12 on UDP `port` 9870, with the shared secret of `X30_TELEOP_TOKEN`.

`max_vx`, `max_vy`, `max_wz` the stick's full deflection, never above the profile's limits (each axis is capped separately): a full stick sideways asks for `max_vy`.

`client_timeout_sec` a handheld silent this long is dropped and its stick zeroed; the mux's `remote_timeout_sec` stops the robot earlier.

`camera_fps`, `camera_jpeg_quality`, `camera_max_width` the video to the handheld: lower them on a weak link before anything else.

`lidar_hz`, `lidar_voxel`, `lidar_max_points` the cloud on the handheld's map.

`patrol_speed_min`, `patrol_speed_max` the range of the patrol speed the app sets (the follower's `cruise`).

11.3 The NETRA uplink

`fleet_uplink_node` (navigation host) connects the robot to NETRA at `server_url` with its `robot_id` and the token of `X30_NETRA_TOKEN`.

`telemetry_hz` snapshots to NETRA (1 Hz).

`spool_dir`, `upload_period_sec` routes and inspections wait on disk until NETRA confirms them, tried every 5 s: a patrol without network loses nothing.

`clock_sync_period_sec`, `clock_warn_ms` the offset to NETRA's clock, measured every 10 s, sent with the telemetry and applied to the times of the uploads; above 200 ms the robot warns: set the robot's clock right (chrony).

11.4 The simulator

`x30_sim` replaces the driver, the LiDARs, the IMU and FAST-LIO with a planar simulation that publishes the same topics: routes, the follower, avoidance, the field and the inspection run unchanged. Its parameters (Appendix A) set the world, the noise and the robot's response; tune the stack's logic there, never its safety distances, which belong to the real robot's measurements.

APPENDIX A

Parameter reference

Every parameter of every node, generated from the configuration declarations of the sources (`docs/tools/params.py`): name, default, unit, accepted range and meaning. The default is the value in the code; the X30's YAML files set some of them for the robot (the profile in `x30_bringup/config/robot.yaml`, every node in its package's `config/`), and those files win. Parameters marked † are read at start only (`live=False`); every other one can be changed while the node runs (`rosparam set`) and is checked before it applies.

A.1 Robot profile

`every node (robot/...)`

Parameter	Default	Unit	Range	Description
`robot.name`	`robot`			robot model, for logs
`robot.kinematics`	`omni`		omni, diff	omni: vx, vy, wz; diff: vx, wz
`robot.base_height`	0.35	m	0.01 .. 3	base_link above the floor when standing
`robot.footprint.front`	0.4	m	0.01 .. 5	base_link to the front edge
`robot.footprint.rear`	0.4	m	0.01 .. 5	base_link to the rear edge
`robot.footprint.half_width`	0.28	m	0.01 .. 5	half of the body width
`robot.limits.max_vx`	0.8	m/s	0 .. 10	forward / backward speed
`robot.limits.max_vy`	0.4	m/s	0 .. 10	sideways speed (omni only)
`robot.limits.max_wz`	1	rad/s	0 .. 10	yaw rate
`robot.limits.max_accel`	0.5	m/s^2	0.01 .. 20	linear acceleration
`robot.limits.max_decel`	0.6	m/s^2	0.05 .. 20	braking deceleration
`robot.limits.max_yaw_accel`	1.5	rad/s^2	0.05 .. 50	yaw braking
`robot.limits.reaction_time`	0.25	s	0 .. 2	command to braking delay (sensing + transport)
`robot.frames.map`	`map`			global, fixed frame
`robot.frames.odom`	`odom`			continuous odometry frame
`robot.frames.base`	`base_link`			robot body frame

A.2 Driver

The driver's parameters come from the comments of `x30_driver/config/driver_params.yaml`.

`x30_driver`

Parameter	Default	Unit	Range	Description
motion_host_ip	192.168.1.103			network.toml (X30 Application Manual 4.3.2; README "Network").
motion_host_port	43893			
local_ip	0.0.0.0			address the report socket binds to
local_port	43897			the port listed in network.toml
heartbeat_hz	5.0			1.2.1: at least 2 Hz (lower values: 2), or the link drops
connect_retry_sec	5.0			resend the connect command this often while no reports come
command_prefix	0x31			automatic mode switch and the takeover check follow the prefix.
save_data_code	0x010C01			emergency stop or fall: the robot exits its motion program.
perception_host_ip	192.168.1.105			Perception host (1.4 autonomous charging, 1.5 elevation map). Empty: no perception services.
elevation_map_port	43899			1.5: commands and the 10 Hz mode feedback
charge_port	3333			1.4: request / response
lio_port	60000			1.5.2.4/5: LiDAR-inertial odometry on / off (multi-frame mode)
obstacle_height_code	0x3100EE04			perception host does not change, try 0x3101EE04 (the other commands are 0x3101EE0x).
elevation_query_sec	5.0			restart the elevation map feedback this often when it stops
charge_query_sec	2.0			ask for the charging state this often (0: never)
feedback_timeout_sec	2.0			how long a service waits for the robot to report the change
velocity_mode	axis			motion program runs / cmd_vel_corrected in non-manual mode, 2.3).
velocity_topic	/cmd_vel_corrected			
cmd_rate_hz	50.0			1.2.3: 50 Hz; the robot drops an axis after 1 s
cmd_timeout_sec	0.5			cmd_vel older than this: zeros
stop_burst_sec	0.5			zeros streamed after motion (at least one), then silence
max_vx	1.0			m/s, clamp of cmd_vel; the gait's own maximum applies too
max_vy	0.5			m/s
max_wz	1.0			rad/s
axis_dead_zone	655			1.2.3.2: |value| < 655 is 0; smaller commands go out as 656
gait_limits				number; listed Appendix B gaits are replaced. Example: {silent: [0.5, 0.5, 0.2, 0.5]}

Parameter	Default	Unit	Range	Description
invert_vy	false			true if +linear.y moves the robot RIGHT (README checklist)
invert_wz	false			true if +angular.z turns the robot RIGHT (README checklist)
auto_control_mode	true			episode start: that mode (fresh report), 3 tries in 2 s
auto_stepping	true			episode start: stepping from torque standing, 3 tries in
idle_stop_stepping_sec	0.0			stop stepping after this long without motion (0: never)
auto_torque_control	true			stand_up continues from initial standing to torque control
stand_timeout_sec	10.0			stand_up / sit_down wait this long for the robot
transition_timeout_sec	5.0			torque control, stepping and e-stop wait this long; also how
odom_frame	odom			
base_frame	base_link			
imu_frame	imu_link			
publish_tf	false			odom -> base_link on /tf (off: the odometry stack owns it)
odom_hz	50.0			one odom per new motion state report, at most this rate
imu_hz	100.0			one imu per new sensor report, at most this rate
joint_state_hz	50.0			
state_hz	10.0			state (diagnostic_msgs/ DiagnosticStatus)
battery_hz	1.0			the robot reports its battery at 0.5 Hz
state_timeout_sec	1.0			no report for this long: disconnected (0x1009: no axes)
low_battery	0.2			0-1, the state warns below
battery_voltage_scale	0.0			V per unit; 0: V, or mV / 10 mV recognised by size
battery_current_scale	0.01			A per unit: 0.01 (10 mA, Deeprcs >= 1.10.19) or 1.0

A.3 Control mux and protective field

control_mux_node

Parameter	Default	Unit	Range	Description
mode†	auto		auto, remote	mode at start
remote_sources†	handheld, dashboard			operator sources, each on control/ remote/<source>/cmd_vel; the first is the default

Parameter	Default	Unit	Range	Description
`nav_timeout_sec`†	0.5	s	0.02 .. 5	zero once when navigation is silent
`remote_timeout_sec`†	0.3	s	0.02 .. 5	zero once when the operator is silent
`estop_hold_hz`†	10	Hz	1 .. 100	zero commands while engaged
`rearm_neutral_count`†	3		1 .. 100	consecutive neutral commands the source must send before it drives again
`neutral_eps`†	0.01	m/s, rad/s	0.0001 .. 0.2	neutral: every axis below
`hw_estop_timeout_sec`†	0	s	0 .. 10	engage when control/hw_estop is silent this long (0: the input is optional)
`use_health_limit`†	`false`			scale every command by health/status speed_scale
`health_timeout_sec`†	3	s	0.1 .. 60	older health/status is ignored
`status_hz`†	2	Hz	0.2 .. 50	control/status period (also on change)

`control_mux_node`: protective field

Parameter	Default	Unit	Range	Description
`field.in_remote`†	`true`			apply the field in REMOTE too (remote_margin)
`field.margin`†	0.15	m	0 .. 2	gap kept to an obstacle in AUTO
`field.remote_margin`†	0.1	m	0 .. 2	gap kept to an obstacle in REMOTE
`field.lateral_margin`†	0.1	m	0 .. 1	extra width of the swept path
`field.human_speed`†	0.5	m/s	0 .. 3	assumed speed of a person walking toward the robot (ISO/TS 15066 separation)
`field.yaw_margin`†	0.1	rad	0 .. 1	turn kept short of a contact
`field.min_speed`†	0.02	m/s	0 .. 0.5	a slower allowed speed stops instead
`field.min_yaw_rate`†	0.05	rad/s	0 .. 1	a slower allowed turn stops instead
`field.remote_degraded_speed`†	0.15	m/s	0 .. 1	REMOTE with a stale or blind scan: fastest point of the body
`field.scan_timeout_sec`†	0.5	s	0.05 .. 5	scan receive age (monotonic) limit
`field.scan_max_age_sec`†	0.5	s	0.05 .. 5	scan header stamp age limit
`field.scan_clock_tolerance_sec`†	0.1	s	0 .. 1	accepted negative stamp age (clocks)

A.4 Perception

`lidar_sector_node`

Parameter	Default	Unit	Range	Description
`input.type`†	`livox`		livox, pointcloud2	livox: raw Livox CustomMsg bytes; pointcloud2: any PointCloud2
`input.livox_topic`†	`livox/lidar`			Livox CustomMsg topic
`input.pointcloud_topic`†	`/cloud_registered_body`			PointCloud2 topic (frame fixed to the robot)
`input.frame`†	`livox_frame`			sensor frame when the header has none
`input.timeout_sec`†	0.5	s	0.05 .. 10	no usable cloud for this long: every sector is published BLOCKED
`front.num_sectors`†	32		1 .. 1440	sectors over the fov
`front.fov_deg`†	180	°	1 .. 360	centred ahead
`front.range_min`†	0.1	m	0 .. 10	closer: ignored
`front.range_max`†	8	m	0.1 .. 100	free beyond
`front.k`†	2		1 .. 100	a sector needs k points; its range is the k-th nearest
`safety.enabled`†	`true`			publish perception/safety_scan
`safety.num_sectors`†	72		4 .. 1440	sectors over 360 deg
`safety.range_min`†	0.1	m	0 .. 10	closer: ignored
`safety.range_max`†	4	m	0.1 .. 100	free beyond
`safety.k`†	2		1 .. 100	a sector needs k points; its range is the k-th nearest
`filter.self_margin`†	0.05	m	0 .. 1	footprint grown by this is the robot
`filter.level_source`†	`tf`		tf, none	tf: roll/pitch of base in odom (unknown -> BLOCKED); none: base assumed level
`filter.tilt_max_age`†	0.2	s	0.01 .. 5	oldest odom->base usable for leveling
`filter.height_min`†	0.1	m	-1 .. 3	band bottom above the floor
`filter.height_max`†	1.6	m	0 .. 5	band top above the floor
`filter.ground_segmentation`†	`false`			estimate the floor height per cell (ramps, uneven floors)
`filter.ground_cell`†	0.3	m	0.05 .. 5	ground segmentation cell size
`filter.ground_max_step`†	0.05	m	0 .. 0.5	largest local floor offset accepted
`filter.drop_adhesion_noise`†	`true`			Livox tag bits 0-1 (glue points)
`filter.drop_particle_noise`†	`true`			Livox tag bits 2-3 (rain, dust)
`filter.drop_other_noise`†	`false`			Livox tag bits 4-5 (other)
`filter.noise_level`†	2		1 .. 2	drop at this tag confidence: 1 medium, 2 low
`memory.enabled`†	`true`			remember obstacles that left the view
`memory.memory_sec`†	10	s	0 .. 60	remembered points decay after this
`memory.radius`†	5	m	0.1 .. 20	points this close are remembered (merged only while below the lowest beam)
`memory.resolution`†	0.05	m	0.01 .. 1	memory cell size
`memory.max_cells`†	20000		100 ..	memory size bound

Parameter	Default	Unit	Range	Description
`memory.lower_fov_deg`†	-7	°	-90 .. 0	lowest beam elevation of the sensor (Mid-360: -7), for the near-field blind zone
`memory.max_jump`†	0.3	m	0 .. 10	odom jump beyond the motion: forget
`blind.min_points`†	100		0 ..	fewer valid points in a cloud: blind
`blind.regions`†	4		1 .. 36	azimuth regions around the sensor z axis
`blind.min_region_points`†	0		0 .. 100000	fewer points in any region: blind (0: off)
`depth.enabled`†	`false`			merge a depth camera cloud into both scans
`depth.topic`†	`/camera/camera/depth/color/points`			PointCloud2 of the depth camera
`depth.frame`†	`camera_depth_optical_frame`			camera frame when the header has none
`depth.timeout_sec`†	0.3	s	0.05 .. 5	an older cloud is not merged: the scans come from the LiDAR alone
`depth.required`†	`false`			no usable depth cloud: every sector BLOCKED (false: LiDAR alone)
`depth.range_max`†	3	m	0.2 .. 20	farther from the camera: dropped (the stereo depth error grows with range squared)
`depth.min_points`†	200		0 ..	fewer points in a cloud (robot body excluded): the camera is blind, not merged
`depth.cell_size`†	0.05	m	0.01 .. 1	obstacle cell edge length
`depth.cell_points`†	4		1 .. 1000	an obstacle needs this many points in one cell (a stereo speckle is one point)
`depth.confirm_radius`†	0.15	m	0 .. 2	an obstacle cell needs one this close in the previous cloud too (0: off)
`depth.memory_sec`†	6	s	0 .. 60	camera obstacles stay this long while out of its view, below the LiDAR (0: off)
`depth.floor_fit`†	`true`			heights above the floor the camera sees (mount, body pitch)
`depth.floor_max_tilt_deg`†	5	°	0 .. 30	a fitted floor tilted more than this from the nominal one is not used
`depth.floor_max_offset`†	0.05	m	0 .. 1	a fitted floor further than this from the nominal one under the robot is not used
`publish_markers`†	`true`			perception/obstacles for RViz

`aruco_detector_node`

Parameter	Default	Unit	Range	Description
`dictionary`†	`DICT_4X4_50`			cv2.aruco predefined dictionary
`ids`†	`4`			marker ids published as TF frames
`frames`†	`charging_dock`			TF frame per id
`sizes`†	0.15	m		edge length per id
`max_distance`	6	m	0.1 .. 50	poses further than this are dropped

A.5 Odometry

`wait_still (FAST-LIO start gate)`

Parameter	Default	Unit	Range	Description
`imu_topic`†	`/imu/data`			IMU topic (sensor_msgs/Imu)
`still_sec`	2	s	0.2 .. 60	this long without motion before the start
`max_gyro_rad_s`	0.03	rad/s	0 .. 10	largest angular rate in the window
`max_acc_rel_std`	0.01		0 .. 1	std of |acc| / mean |acc| (unit-free: m/s^2 or g)
`max_tilt_deg`	3	°	0 .. 90	gravity against the IMU z axis; 0: off
`min_samples`	20		2 .. 10000	IMU samples needed in the window
`timeout_sec`	60	s	0 .. 3600	start anyway (with an error); 0: wait forever

`odometry_monitor_node`

Parameter	Default	Unit	Range	Description
`odom_topic`†	`/Odometry`			estimator odometry (nav_msgs/Odometry)
`leg_odom_topic`†	`""`			robot leg / wheel odometry for the speed test; empty: off
`imu_topic`†	`/imu/data`			IMU with the axes of the odometry body; empty: no tilt
`status_period_sec`†	0.2	s	0.01 .. 10	odometry/status period
`stale_sec`	0.5	s	0.05 .. 60	no odometry this long: LOST
`hold_sec`	5	s	0 .. 600	a jump or restart stays LOST this long
`max_z_drift_m`	0.5	m	0 .. 100	|z| above this is DEGRADED (flat floor); 0: off
`restart.gap_sec`	2	s	0.1 .. 60	a message gap this long followed by the origin
`restart.origin_radius_m`	0.05	m	0 .. 1	a fresh estimator starts this close to the origin
`restart.origin_yaw_deg`	1	°	0 .. 180	and with this little yaw
`jump.speed_factor`	3		1 .. 100	a step faster than this times the robot speed limit
`jump.min_step_m`	0.3	m	0 .. 100	and longer than this is a jump

Parameter	Default	Unit	Range	Description
speed.window_sec	0.5	s	0.05 .. 10	estimator speed from the poses of this window
speed.sigma_lio_mps	0.05	m/s	0.001 .. 10	std of the estimator speed
speed.sigma_leg_mps	0.15	m/s	0.001 .. 10	std of the leg / wheel odometry speed
speed.cusum_drift	3.84		0 .. 1000	kappa, subtracted from every chi^2 (chi^2(1) 95 %: 3.84)
speed.cusum_threshold	20		0.1 .. 10000	h: the CUSUM above this is a mismatch
speed.leg_timeout_sec	0.3	s	0.01 .. 10	older leg odometry is not compared
tilt.max_deg	5	°	0 .. 90	odom z against gravity above this; 0: off
tilt.still_sec	1	s	0.2 .. 60	IMU window
tilt.max_gyro_rad_s	0.03	rad/s	0 .. 10	still: rate
tilt.max_acc_rel_std	0.01		0 .. 1	still: std of |acc| / mean
tilt.min_samples	20		2 .. 10000	IMU samples needed in the window
trip.enabled†	true			odometry/trip and odometry/trip_reset
trip.file†	~/.x30/trip.json			totals kept across restarts; empty: not kept
trip.step_m	0.05	m	0.001 .. 1	distance grows in steps of this (pose noise: 0)
trip.window_sec	1	s	0.1 .. 10	the current speed over this window
trip.moving_mps	0.05	m/s	0 .. 1	slower counts as standing (moving time, speed 0)
trip.gap_sec	1	s	0.1 .. 60	a longer pose gap counts nothing across it
trip.max_mps	3	m/s	0.1 .. 50	a faster step is a jump, not travelled
trip.period_sec†	0.5	s	0.05 .. 10	odometry/trip period
trip.save_sec†	10	s	1 .. 3600	totals written at most this often

A.6 Localization

localization_node

Parameter	Default	Unit	Range	Description
sites_dir†	~/x30/map			load_tags paths must be inside
tags_file†	""			tags file at start (empty: none until load_tags)
lio_odom_frame†	camera_init			LIO world frame
lio_body_frame†	body			LIO body frame (the IMU)
imu_frame†	imu_link			URDF frame of the LIO body
publish_hz†	20	Hz	1 .. 200	map->odom on /tf
pose_hz†	10	Hz	0 .. 100	localization/pose; 0: off
assume_start_origin†	true			the odom origin is the map origin at start (no tag needed)

Parameter	Default	Unit	Range	Description
estimator.sigma_xy_per_sqrt_m	0.03	m/$\sqrt{\text{m}}$	0 .. 1	position drift: 1-sigma after 1 m of travel
estimator.sigma_yaw_deg_per_sqrt_m	0.1	°/$\sqrt{\text{m}}$	0 .. 10	heading drift per metre of travel
estimator.sigma_yaw_deg_per_sqrt_rad	0.5	°/$\sqrt{\text{rad}}$	0 .. 10	heading drift per radian turned
estimator.gate_confidence	0.99		0.95, 0.99, 0.999	chi-square gate (3 dof) of automatic corrections
estimator.max_sigma_xy	100	m	0.01 .. 100	above: state UNCERTAIN (position)
estimator.max_sigma_yaw_deg	180	°	0.1 .. 180	above: state UNCERTAIN (heading)
estimator.origin_sigma_xy	0.05	m	0 .. 1	position of the start spot (assume_origin)
estimator.origin_sigma_yaw_deg	2	°	0 .. 45	heading on the start spot (assume_origin)
estimator.min_sigma_xy	0.01	m	0 .. 1	floor after a correction (no overconfidence)
estimator.min_sigma_yaw_deg	0.25	°	0 .. 10	heading floor after a correction
estimator.motion_step_m	0.05	m	0.001 .. 1	odometry motion counted in steps (jitter-free)
estimator.motion_step_deg	2	°	0.01 .. 45	odometry rotation counted in steps
tag.camera_frame†	camera_color_optical_frame			frame the detector puts the tags under
tag.frames	base_map, charging_dock			tag TF frames to record (the detector tag frames)
tag.max_age_sec	0.5	s	0.01 .. 10	older detections ignored
tag.max_distance_m	4	m	0.1 .. 50	farther tags ignored
tag.max_view_angle_deg	60	°	1 .. 89	tags seen more obliquely are ignored
tag.sigma_xy_at_1m	0.02	m	0.001 .. 1	position sigma at 1 m, grows with distance^2
tag.sigma_yaw_deg_at_1m	1	°	0.01 .. 30	heading sigma at 1 m, grows with distance
tag.update_hz†	10	Hz	0.5 .. 50	tag checks
tag.min_interval_sec	0.5	s	0 .. 60	per tag, between two fused detections
tag.auto_localize	true			while not localized: localize from any visible tag
tag.auto_correct	true			while localized: fuse visible tags through the gate
tag.record_duration_sec	2	s	0.2 .. 60	record_tags
tag.record_min_samples	5		1 .. 1000	detections needed per tag
tag.record_max_std_m	0.05	m	0.001 .. 1	noisier recordings are refused
tag.record_max_std_deg	3	°	0.01 .. 45	noisier recordings are refused (heading)
tag.marker_size	0.15	m	0.01 .. 2	tag plate in RViz

Parameter	Default	Unit	Range	Description
`leveling.enabled`†	`true`			level odom with the IMU (off: odom = base)
`leveling.imu_topic`†	`/imu/data`			IMU in the LIO body frame axes
`leveling.window_sec`	1	s	0.1 .. 10	still window
`leveling.max_rate_deg_s`	3	°/s	0.1 .. 90	still: gyro rate below this
`leveling.max_accel_spread`	0.03		0.001 .. 0.5	still: accelerometer std / mean norm below this
`leveling.max_tilt_deg`	20	°	1 .. 60	larger start tilts are refused (wrong frame?)
`leveling.timeout_sec`	15	s	0 .. 600	then the bridge goes up unleveled; 0: wait

A.7 Navigation

`site_manager_node`

Parameter	Default	Unit	Range	Description
`sites_dir`†	`~/x30/map`			one directory per site
`site`†	`""`			site loaded at start; empty: the one of the last run
`state_file`†	`~/.x30/site`			remembers the active site
`create_site`†	`true`			create 'site' at start when missing: its routes and tags
`service_timeout_sec`†	5	s	0.5 .. 60	waiting for localization
`refresh_sec`†	2	s	0.5 .. 60	site/status refresh (tags, routes)

`route_recorder_node`

Parameter	Default	Unit	Range	Description
`sites_dir`†	`~/x30/map`			routes go to <sites_dir>/<site>/routes
`output_path`†	`""`			directory for routes while no site is active; empty: refuse
`min_spacing_m`	0.2	m	0.01 .. 10	a new waypoint after this much motion
`record_rate_hz`†	10	Hz	0.5 .. 100	pose sampling while recording
`flush_sec`	2	s	0.1 .. 60	the route file is rewritten at most this often
`max_tf_age_sec`	0.5	s	0.01 .. 10	older map->base: no sample

`path_loader_node`

Parameter	Default	Unit	Range	Description
`sites_dir`†	`~/x30/map`			routes are loaded from inside this directory only
`route`†	`""`			route CSV loaded at start (inside sites_dir)
`remember_file`†	`~/.x30/last_route`			last loaded route, reloaded at start (“”: off)
`process`	`true`			resample and smooth routes between stop points
`processing.spacing`	0.1	m	0.02 .. 1	point spacing after resampling
`processing.smooth_weight_data`	0.05		0 .. 1	pull toward the recording
`processing.smooth_weight_smooth`	0.4		0 .. 0.49	pull toward neighbours
`processing.max_deviation`	0.3	m	0 .. 5	warn when smoothing moves the route further
`republish_sec`	2	s	0 .. 60	navigation/path republish for RViz (0: off)

`path_follower_node`

Parameter	Default	Unit	Range	Description
`control_rate`†	20	Hz	1 .. 200	loop rate
`controller`	`holonomic`			path controller (x30.controllers)
`avoidance_mode`	`path_shift`		path_shift, reactive, off	path_shift: slide sideways (holonomic)
`loop_route`	`false`			restart a closed route at its end (patrol)
`loop_close_distance`	1	m	0 .. 20	end this close to the start: a closed route
`require_localization`	`true`			pause while localization/status is NONE
`pose_wait_sec`	2	s	0 .. 30	navigation/start waits this long for the robot pose before refusing
`control_node`†	`control_mux_node`			control mux whose field the guard must cover (checked once; empty: no check)
`low_battery`	0.2		0 .. 1	below: finish the lap, no new one
`critical_battery`	0.1		0 .. 1	below: pause (reason battery)
`battery_hysteresis`	0.05		0 .. 0.5	recharge above + this
`speed.cruise`	0.4	m/s	0.01 .. 5	cruise speed
`speed.min_speed`	0.1	m/s	0 .. 5	floor of the curve limit
`speed.approach_speed`	0.15	m/s	0.01 .. 5	speed at stop points and the route end
`speed.lateral_accel`	0.3	m/s^2	0.01 .. 10	curve limit v = sqrt(a / curvature)
`speed.accel`	0.3	m/s^2	0.01 .. 20	comfortable acceleration
`speed.decel`	0.3	m/s^2	0.01 .. 20	comfortable braking

Parameter	Default	Unit	Range	Description
speed.smoothing	0.5		0.05 .. 1	weight of a new yaw rate / sideways command (1: off)
safety.stop_margin†	0.3	m	0 .. 5	gap kept to obstacles when stopping
safety.yaw_margin_deg†	6	°	0 .. 90	turning room kept to obstacles, >= the control field.yaw_margin (0.10 rad)
safety.scan_timeout_sec†	0.5	s	0.05 .. 5	older safety scans stop the robot
safety.avoidance_timeout_sec†	0.5	s	0.05 .. 5	older avoidance (reactive) stops it
safety.tf_timeout_sec†	0.5	s	0.05 .. 5	an older pose stops the robot
safety.tf_compensate_sec†	0.1	s	0 .. 5	older poses are predicted forward
safety.clock_tolerance_sec†	0.1	s	0 .. 5	a pose from further in the future: clock
safety.emergency_release_sec	1	s	0 .. 30	an emergency stop holds this long after it clears
safety.lateral_margin†	0.1	m	0 .. 1	extra width of the guarded path, >= the control field.lateral_margin
safety.human_speed†	0.5	m/s	0 .. 3	assumed speed of a person walking toward the robot, >= the control field.human_speed
safety.pass_margin†	0.15	m	0 .. 5	gap kept to a point passed within lateral_margin, >= the control field.margin
safety.blocked_timeout_sec	10	s	0.5 .. 600	standing still this long: BLOCKED
safety.stall_distance	0.1	m	0 .. 2	moved less than this (measured): standing still
safety.stall_yaw_deg	10	°	0 .. 180	turned less than this: standing still
safety.off_path_slow	0.6	m	0.05 .. 10	further from the route: slow to the floor speed
safety.off_path_stop	1.5	m	0.1 .. 20	further from the route: OFF_PATH, stopped
path_shift.max_offset	0.8	m	0 .. 5	furthest from the route
path_shift.step	0.1	m	0.02 .. 1	candidate offset spacing
path_shift.look_ahead	2	m	0.1 .. 20	route ahead of the front edge that must be free
path_shift.clearance	0.1	m	0 .. 2	gap kept beside the body
path_shift.offset_rate	0.3	m/s	0.01 .. 5	how fast it slides onto a new offset
path_shift.max_lead	0.25	m	0.02 .. 2	the offset leads the robot by at most this
path_shift.switch_cost	0.5		0 .. 10	keeps its side (hysteresis)
path_shift.switch_margin	0.05	m	0 .. 1	extra clearance a new offset needs
path_shift.return_delay_sec	0.5	s	0 .. 30	a way back must stay proposed this long
path_shift.stop_margin	0.4	m	0 .. 5	stop this far before an obstacle (front edge)

Parameter	Default	Unit	Range	Description
path_shift.patience_sec	3	s	0 .. 600	held this long with no clear offset: search
path_shift.search_offset	1.5	m	0 .. 5	furthest while searching (<= max_offset: off)
path_shift.emergency_slide	true			an emergency holds the forward motion only: the robot still slides out
path_shift.centre_room	0.25	m	0 .. 2	a lane with less room beside the body is driven in its middle (<= clearance: off)
detour.enabled	true			turn toward a gap and drive through it when stuck
detour.after_sec	1.5	s	0.5 .. 600	stuck this long while following: detour
detour.max_angle_deg	75	°	5 .. 135	headings this far off the route direction
detour.length	2	m	0.3 .. 10	longest straight run
detour.min_progress	0.3	m	0 .. 5	a run must take the robot this far along the route
detour.speed	0.25	m/s	0.05 .. 1	speed on the run
backtrack.enabled	true			back up along the driven path when stuck, then try again
backtrack.after_sec	30	s	5 .. 600	no progress along the route this long: back up
backtrack.min_progress	0.3	m	0.05 .. 5	less progress than this counts as no progress
backtrack.distance	1	m	0.2 .. 5	how far back along the driven path
backtrack.speed	0.15	m/s	0.05 .. 0.5	backing-up speed (360 deg guard, rear sectors)
backtrack.max_attempts	3		1 .. 20	attempts at one place, then it stays BLOCKED (waits)
approach.radius	0.5	m	0.05 .. 5	this close (along the route) the approach starts
approach.skip_margin	0.5	m	0.05 .. 10	progress this far past a stop: skipped
approach.timeout_sec	20	s	1 .. 600	give up converging (paused time not counted)
approach.kp_xy	1.2	1/s	0 .. 10	gain on the position error
approach.kp_yaw	1.5	1/s	0 .. 10	gain on the yaw error
approach.max_speed	0.2	m/s	0.01 .. 2	approach speed cap
approach.max_yaw_rate	0.6	rad/s	0.01 .. 5	yaw rate cap
approach.min_speed	0.04	m/s	0 .. 1	floor to beat the robot deadband
approach.position_tolerance	0.05	m	0.005 .. 1	arrived within
approach.yaw_tolerance_deg	3	°	0.1 .. 45	aligned within
join.max_distance	3	m	0 .. 50	a robot this far walks onto the route (0: off)
join.timeout_sec	60	s	1 .. 600	give up and follow from where it is
join.on_route	0.15	m	0.01 .. 2	joined within this

Parameter	Default	Unit	Range	Description
`hooks.start`†	`[]`			called on start; the robot waits for every success
`hooks.stop`†	`[]`			called on stop (not waited for)
`hooks.inspection`†	`[]`			called at every stop point
`hooks.timeout_sec`†	5	s	0.1 .. 120	start and stop hook timeout
`hooks.inspection_timeout_sec`	20	s	0.1 .. 600	a stop waits this long for inspections
`inspection.reverse_distance_m`	0	m	0 .. 5	back up this far after the dwell (0: off)
`inspection.reverse_speed`	0.1	m/s	0.01 .. 1	backing-up speed (360 deg guard, rear sectors)
`regulator.enabled`	`false`			scale the speed down when turning hard
`regulator.reference_yaw_rate`	0.3	rad/s	0 .. 5	turning slower than this needs no slow-down
`regulator.kp`	0.5	s/rad	0 .. 10	scale per rad/s above it
`regulator.ki`	0	1/rad	0 .. 10	scale per integrated rad
`regulator.integral_limit`	1	rad	0 .. 10	anti-windup clamp
`regulator.min_scale`	0.3		0 .. 1	never slower than this fraction
`processing.spacing`	0.1	m	0.02 .. 1	point spacing after resampling
`processing.smooth_weight_data`	0.05		0 .. 1	pull toward the recording
`processing.smooth_weight_smooth`	0.4		0 .. 0.49	pull toward neighbours
`processing.max_deviation`	0.3	m	0 .. 5	warn when smoothing moves the route further
`markers`	`true`			publish RViz markers

`obstacle_avoidance_node`

Parameter	Default	Unit	Range	Description
`policy`	`vfh`			avoidance policy (x30.avoidance)
`shared.safe_distance`	1.5	m	0.1 .. 20	obstacles closer than this shape the command (beyond the guard stopping reach)
`shared.emergency_distance`	0.75	m	0 .. 10	obstacle in the cone closer than this: stop
`shared.emergency_cone_deg`	60	°	1 .. 360	frontal cone of the emergency stop
`shared.max_heading_offset_deg`	60	°	0 .. 180	largest turn of the direction of travel
`shared.smoothing`	0.8		0 .. 0.99	0..1 filter on steering and speeding up (never on braking)
`shared.window_deg`	180	°	10 .. 360	sectors evaluated around the travel direction

Parameter	Default	Unit	Range	Description
shared.body_margin	0.15	m	0 .. 2	gap kept beside the body on a chosen heading (> the guard lateral_margin)
braitenberg.steering_gain	1	rad	0 .. 10	turn per unit danger
braitenberg.velocity_gain	1		0 .. 10	slow-down per unit danger
braitenberg.steering_zone_center_deg	45	°	0 .. 180	
braitenberg.steering_zone_width_deg	60	°	1 .. 360	
braitenberg.velocity_zone_center_deg	60	°	0 .. 180	
braitenberg.velocity_zone_width_deg	90	°	1 .. 360	
braitenberg.following_zone_width_deg	30	°	1 .. 360	
vfh.min_gap_width_deg	20	°	1 .. 180	narrowest run of free headings taken
vfh.steering_gain	1		0 .. 5	fraction of the gap heading taken
vfh.velocity_gain	1		0 .. 10	slow-down per unit frontal danger
potential_field.repulsion_gain	0.5		0 .. 10	strength of the repulsion
potential_field.steering_gain	0.8	rad	0 .. 5	turn per atan(force)
potential_field.lateral_gain	0.3		0 .. 5	sideways m/s per unit force
potential_field.max_lateral	0.2	m/s	0 .. 2	cap on the sideways push
potential_field.velocity_gain	1		0 .. 10	slow-down per unit frontal danger
follow_gap.bubble_radius	0.45	m	0 .. 5	safety bubble around the nearest obstacle (at least half_width + body_margin)
follow_gap.steering_gain	1		0 .. 5	fraction of the gap heading taken
follow_gap.velocity_gain	1		0 .. 10	slow-down per unit frontal danger
markers	true			publish RViz markers

A.8 Path controllers

path_follower_node: holonomic

Parameter	Default	Unit	Range	Description
holonomic.lookahead	0.6	m	0.1 .. 5	carrot ahead along the route; longer = gentler

Parameter	Default	Unit	Range	Description
holonomic.heading_lookahead	0.6	m	0.1 .. 5	the heading follows the route this far ahead
holonomic.kp_yaw	1.5	1/s	0 .. 10	yaw rate per rad of error
holonomic.max_lateral_speed	0.25	m/s	0 .. 5	sideways speed cap (0: never strafe)
holonomic.max_reverse_speed	0.15	m/s	0 .. 5	backward speed cap when the route is behind
holonomic.min_alignment_scale	0.25		0 .. 1	translation factor while the heading is far off

path_follower_node: pure_pursuit

Parameter	Default	Unit	Range	Description
pure_pursuit.lookahead	1	m	0.1 .. 5	carrot ahead along the route

path_follower_node: pid

Parameter	Default	Unit	Range	Description
pid.lookahead	1	m	0.1 .. 5	carrot ahead along the route
pid.kp	1.5	1/s	0 .. 20	proportional gain
pid.ki	0	$1/s^2$	0 .. 20	integral gain
pid.kd	0.2		0 .. 20	derivative gain
pid.integral_limit	0.5	rad s	0 .. 10	anti-windup clamp of the error integral

path_follower_node: lqr

Parameter	Default	Unit	Range	Description
lqr.q_lateral	1		0 .. 100	weight on the cross-track error
lqr.q_heading	0.5		0 .. 100	weight on the heading error
lqr.r_yaw_rate	0.5		0.001 .. 100	weight on the yaw rate (higher = gentler)

path_follower_node: stanley

Parameter	Default	Unit	Range	Description
stanley.k_cross_track	1		0 .. 20	cross-track gain
stanley.k_soft	0.2	m/s	0.01 .. 5	softening so low speeds do not over-steer
stanley.k_heading	1.5	1/s	0 .. 20	steering to yaw rate gain

`path_follower_node`: mppi

Parameter	Default	Unit	Range	Description
`mppi.horizon_steps`	15		2 .. 100	steps per rollout
`mppi.step_sec`	0.1	s	0.01 .. 1	rollout step
`mppi.samples`	200		10 .. 5000	rollouts per tick
`mppi.yaw_rate_std`	1	rad/s	0.01 .. 5	sampling noise
`mppi.temperature`	0.05		0.001 .. 100	softmax sharpness of the normalized cost
`mppi.window`	4	m	0.5 .. 20	route ahead in the cost
`mppi.obstacle_weight`	5		0 .. 1000	cost per m of intrusion
`mppi.clearance`	0.5	m	0 .. 5	rollouts closer to an obstacle pay for it
`mppi.seed`†	0		0 ..	random seed (reproducible)

`path_follower_node`: neupan

Parameter	Default	Unit	Range	Description
`neupan.dune_checkpoint`†	`""`			DUNE model of the footprint (.pth, its .json beside it)
`neupan.horizon_steps`†	10		2 .. 50	MPC steps ahead
`neupan.step_sec`†	0.1	s	0.02 .. 0.5	MPC step, also the time between solves
`neupan.iterations`†	2		1 .. 10	PAN iterations per solve
`neupan.dune_points`†	100		1 .. 2000	obstacle points through DUNE (decimated beyond)
`neupan.nrmp_points`†	10		1 .. 100	nearest points in the MPC
`neupan.obstacle_range`†	4	m	0.5 .. 50	points further from base_link are left out (inside the model's training range)
`neupan.window`†	5	m	0.5 .. 50	route ahead given
`neupan.spacing`†	0.1	m	0.02 .. 1	between the route points given
`neupan.heading_lookahead`†	0.6	m	0 .. 5	the heading follows the route this far
`neupan.weight_xy`†	1		0 .. 100	position error cost
`neupan.weight_yaw`†	0.5		0 .. 100	heading error cost
`neupan.weight_speed`†	2		0 .. 100	speed error cost: below 2 it stands before obstacles
`neupan.weight_lateral`†	1		0 .. 100	cost of walking sideways (0: free)
`neupan.clearance_gain`†	10		0 .. 1000	reward per m of clearance up to max_clearance
`neupan.max_clearance`†	0.25	m	0 .. 5	clearance rewarded up to here: passages narrower than the body plus twice it stall
`neupan.min_clearance`†	0.1	m	0 .. 2	clearance kept from every point
`neupan.stop_distance`†	0.05	m	0 .. 1	a point this near the body: stand
`neupan.max_reverse_speed`†	0.15	m/s	0 .. 5	backward speed cap

Parameter	Default	Unit	Range	Description
`neupan.max_failures`†	10		1 .. 1000	failed solves in a row before giving up (stop)
`neupan.threads`†	1		0 .. 64	torch threads (0: torch default)

A.9 Reactive avoidance

`obstacle_avoidance_node`: braitenberg

Parameter	Default	Unit	Range	Description
`braitenberg.steering_gain`	1	rad	0 .. 10	turn per unit danger
`braitenberg.velocity_gain`	1		0 .. 10	slow-down per unit danger
`braitenberg.steering_zone_center_deg`	45	°	0 .. 180	
`braitenberg.steering_zone_width_deg`	60	°	1 .. 360	
`braitenberg.velocity_zone_center_deg`	60	°	0 .. 180	
`braitenberg.velocity_zone_width_deg`	90	°	1 .. 360	
`braitenberg.following_zone_width_deg`	30	°	1 .. 360	

`obstacle_avoidance_node`: follow_gap

Parameter	Default	Unit	Range	Description
`follow_gap.bubble_radius`	0.45	m	0 .. 5	safety bubble around the nearest obstacle (at least half_width + body_margin)
`follow_gap.steering_gain`	1		0 .. 5	fraction of the gap heading taken
`follow_gap.velocity_gain`	1		0 .. 10	slow-down per unit frontal danger

`obstacle_avoidance_node`: potential_field

Parameter	Default	Unit	Range	Description
`potential_field.repulsion_gain`	0.5		0 .. 10	strength of the repulsion
`potential_field.steering_gain`	0.8	rad	0 .. 5	turn per atan(force)
`potential_field.lateral_gain`	0.3		0 .. 5	sideways m/s per unit force

Parameter	Default	Unit	Range	Description
potential_field.max_lateral	0.2	m/s	0 .. 2	cap on the sideways push
potential_field.velocity_gain	1		0 .. 10	slow-down per unit frontal danger

obstacle_avoidance_node: vfh

Parameter	Default	Unit	Range	Description
vfh.min_gap_width_deg	20	°	1 .. 180	narrowest run of free headings taken
vfh.steering_gain	1		0 .. 5	fraction of the gap heading taken
vfh.velocity_gain	1		0 .. 10	slow-down per unit frontal danger

A.10 Docking

docking_node

Parameter	Default	Unit	Range	Description
control_rate†	20	Hz	1 .. 200	loop rate
control_node†	control_mux_node			mux whose field is checked against the goal ("": no)
tag.frame†	charging_dock			TF frame of the dock tag
tag.normal_axis	z		x, y, z	tag axis along the plate normal (its sign is found)
tag.timeout_sec	0.5	s	0.05 .. 5	an older detection is not in view
tag.max_range	4	m	0.2 .. 20	further detections ignored
tag.smoothing	0.3		0.01 .. 1	weight of a new detection in the dock estimate (1: none)
tag.max_jump	0.25	m	0.01 .. 5	a detection moving the dock further: outlier
tag.max_outliers	5		1 .. 100	this many outliers in a row: start the estimate over
goal.distance	0.75	m	0.05 .. 5	base_link to the tag along its normal
goal.lateral_offset	0	m	-2 .. 2	goal beside the tag centre (+left facing it)
goal.yaw_offset_deg	0	°	-45 .. 45	goal heading beside the tag normal
goal.direction	forward		forward, backward	forward: faces the tag; backward: backs in
centering.lateral_tolerance	0.03	m	0.002 .. 0.5	centred within this beside the line
centering.yaw_tolerance_deg	2	°	0.1 .. 30	aligned within this to the line
centering.realign_lateral	0.1	m	0.005 .. 1	further beside the line: stop and centre

Parameter	Default	Unit	Range	Description
centering.realign_yaw_deg	8	°	0.5 .. 60	further off the heading: stop and centre
centering.hold_sec	0.3	s	0 .. 10	centred this long before walking on
centering.timeout_sec	30	s	1 .. 600	give up centring (an attempt fails)
approach.speed	0.12	m/s	0.01 .. 1	speed toward the dock
approach.min_speed	0.03	m/s	0 .. 0.5	floor to beat the robot deadband
approach.max_lateral_speed	0.08	m/s	0 .. 1	sideways correction cap (omni)
approach.max_yaw_rate	0.4	rad/s	0.01 .. 3	yaw rate cap
approach.kp_distance	0.8	1/s	0 .. 10	gain on the distance
approach.kp_lateral	1.5	1/s	0 .. 10	gain beside the line
approach.kp_yaw	1.5	1/s	0 .. 10	gain on the heading
approach.look_ahead	0.3	m	0.05 .. 3	steering look-ahead on the line (diff)
approach.blind_distance	0.25	m	0 .. 3	this close the dock estimate freezes and the tag may leave the view
approach.position_tolerance	0.02	m	0.002 .. 0.5	arrived within this of the goal distance
approach.timeout_sec	60	s	1 .. 600	give up the approach (an attempt fails)
settle.time_sec	1	s	0 .. 30	stand still this long
settle.distance_tolerance	0.04	m	0.002 .. 0.5	docked within this along the normal
settle.lateral_tolerance	0.03	m	0.002 .. 0.5	docked within this beside the line
settle.yaw_tolerance_deg	4	°	0.1 .. 45	docked within this heading
retry.max_retries	2		0 .. 20	attempts after the first (0: none)
retry.backoff_distance	0.4	m	0 .. 3	move this far away from the dock first
retry.speed	0.1	m/s	0.01 .. 1	backing-off speed
supervision.search_timeout_sec	5	s	0.1 .. 600	no dock tag this long after a start: failed
supervision.lost_timeout_sec	3	s	0.1 .. 60	tag gone this long beyond blind_distance
supervision.pose_timeout_sec†	0.5	s	0.05 .. 5	an older odom pose stops the robot
supervision.stall_timeout_sec	8	s	0.5 .. 120	commanded but not moving this long: held
supervision.stall_distance	0.02	m	0 .. 1	moved less than this: not moving
supervision.stall_yaw_deg	2	°	0 .. 90	turned less than this: not moving
hooks.sit†	[]			called once docked (e.g. /robot/sit)
hooks.charge†	[]			called after sitting (e.g. /robot/charge/start)
hooks.stop_charge†	[]			called first when undocking (e.g. /robot/charge/stop)
hooks.stand†	[]			called before backing off the dock (e.g. /robot/stand)

Parameter	Default	Unit	Range	Description
`hooks.navigation_stop`†	`navigation/stop`			stops the path follower when docking starts (“”: no)
`hooks.timeout_sec`†	15	s	0.1 .. 600	each hook call times out
`hooks.sit_settle_sec`	1.5	s	0 .. 60	wait after sitting before the charge hooks
`hooks.stand_settle_sec`	1.5	s	0 .. 60	wait after standing before backing off
`charge.confirm_timeout_sec`	30	s	0 .. 600	wait for the battery to charge (0: do not)
`charge.battery_timeout_sec`	5	s	0.1 .. 120	an older battery state counts as unknown
`undock.distance`	0.5	m	0 .. 5	move this far away from the dock (0: stay)
`undock.speed`	0.1	m/s	0.01 .. 1	undocking speed
`undock.timeout_sec`	30	s	1 .. 600	give up undocking

A.11 Health

`health_monitor_node`

Parameter	Default	Unit	Range	Description
`period_sec`†	1	s	0.1 .. 10	health/status and diagnostics period
`startup_grace_sec`	30	s	0 .. 600	missing nodes and inputs only warn this long
`diagnostics_prefix`	`health`			“<prefix>: <check>” in rqt_robot_monitor
`extra_checks`†	`[]`			registered checks to add (entry points x30_health.checks)
`sensors.topics`†	`/livox/lidar:5.0, /Odometry:8.0`			“topic:min_hz[:max_age_sec]” (max_age_sec 0: no stamp check)
`sensors.rate_tolerance`	0.8		0 .. 1	a rate below this share of min_hz is an issue
`sensors.window_sec`†	5	s	0.5 .. 60	rate window
`sensors.max_age_sec`	0.5	s	0 .. 60	stamp age limit of entries without one; 0: off
`sensors.max_future_sec`	0.1	s	0 .. 60	stamps further ahead of the clock: clock offset
`sensors.discover_period_sec`†	2	s	0.1 .. 60	look for unsubscribed topics
`system.thermal_zones`	`cpu*, gpu*, tj*, x86_pkg_temp`			thermal zone types read (fnmatch, any case); others are ignored, some are bogus
`system.max_temp_c`	85	°reeC	0 .. 150	hottest zone
`system.max_memory_percent`	92	%	0 .. 100	memory in use
`system.min_disk_free_gb`	5	GB	0 ..	free at disk_path

Parameter	Default	Unit	Range	Description
system.disk_path	~			file system to watch (~ and $VAR expanded)
system.max_load_per_cpu	2		0 .. 100	1 min load average / CPUs
system.max_cpu_percent	0	%	0 .. 100	CPU busy share
system.max_gpu_percent	0	%	0 .. 100	Jetson GPU load
nodes.expected	[]			"/namespace/name" of every node that must run
state.odometry_topic†	odometry/status			OdometryStatus
state.localization_topic†	localization/status			LocalizationStatus
state.control_topic†	control/status			ControlStatus
state.safety_scan_topic†	perception/safety_scan			SectorScan (360)
state.timeout_sec	3	s	0.1 .. 600	a status older than this is stale
state.scan_timeout_sec	1	s	0.05 .. 60	a safety scan older than this is stale
tf.enabled†	true			listen to TF
tf.edges	[]			"parent->child" dynamic edges that must exist (every dynamic edge is watched)
tf.max_age_sec	0.5	s	0.01 .. 60	newest transform age
tf.max_future_sec	0.1	s	0 .. 60	stamps further ahead of the clock: clock offset
tf.window_sec	5	s	1 .. 60	age history per edge
tf.offset_jitter_sec	0.2	s	0 .. 10	late ages steadier than this: a clock offset
clock.enabled†	false			run the command
clock.command†	"chronyc tracking"			prints "System time : x seconds"
clock.period_sec	10	s	1 .. 3600	run it this often
clock.timeout_sec	1	s	0.1 .. 10	command timeout
clock.max_offset_sec	0.05	s	0 .. 60	|offset| limit
degradation.rules	sensor:/livox/lidar=0.5, sensor:/Odometry=0.5, tf:*=0.5			"<issue name pattern>=<scale>" for warnings (fnmatch); the lowest match wins
degradation.warn_scale	1		0 .. 1	scale of warnings without a rule

A.12 Inspection

inspection_node

Parameter	Default	Unit	Range	Description
enabled	true			false: stop points pass without inspection

Parameter	Default	Unit	Range	Description
`sites_dir`†	`~/x30/map`			<site>/inspection.yaml holds the points and targets
`reports_dir`†	`~/x30/reports`			one folder per inspected point
`reports_keep_days`	0	d	0 .. 36500	days of reports kept; 0: until the disk needs room
`reports_min_free_gb`	6	GB	0 ..	the oldest days of reports go while less is free (x30_health warns at 5); 0: never
`match_radius_m`	1.5	m	0.1 .. 10	a stop this close to a point inspects it
`max_duration_sec`	150	s	5 .. 1800	one point; later targets are skipped
`stop_max_age_sec`	60	s	1 .. 3600	a stop point event older than this is stale
`camera.namespace`†	`/hikmicro`			hikmicro_driver namespace
`camera.move_timeout_sec`	20	s	1 .. 120	a move to a target: arrived or failed by then
`camera.call_timeout_sec`	5	s	0.1 .. 60	each temperature or snapshot call
`camera.settle_sec`	1	s	0 .. 30	after arriving, before measuring (focus, image)
`camera.park_preset`	0		0 .. 300	camera preset called after a point; 0: the camera stays
`camera.state_max_age_sec`	3	s	0.1 .. 60	ptz/state older than this: not connected
`camera.raw_thermal`	`true`			a raw thermometry frame (.npz, degC per pixel) per target in the report
`camera.raw_wait_sec`	2.5	s	0 .. 30	for a frame of thermal/temperature once settled
`thermal.ambient_c`	30	°reeC	-40 .. 60	air temperature at the site (live: set it)
`thermal.samples`	3		1 .. 20	measurements per region; the median is used
`thermal.sample_interval_sec`	0.3	s	0 .. 5	between two measurements
`thermal.max_age_sec`	0.5	s	0.05 .. 5	a thermometry frame at most this old per sample
`thermal.absolute.attention`	80	°reeC	0 .. 500	possible deficiency
`thermal.absolute.warning`	90	°reeC	0 .. 500	probable deficiency
`thermal.absolute.critical`	105	°reeC	0 .. 500	major deficiency
`thermal.ambient.attention`	11	°reeC	0 .. 500	possible deficiency
`thermal.ambient.warning`	21	°reeC	0 .. 500	probable deficiency
`thermal.ambient.critical`	41	°reeC	0 .. 500	major deficiency
`thermal.similar.attention`	2	°reeC	0 .. 500	possible deficiency
`thermal.similar.warning`	4	°reeC	0 .. 500	probable deficiency

Parameter	Default	Unit	Range	Description
thermal.similar.critical	16	°reeC	0 .. 500	major deficiency
acoustic.enabled	true			false: acoustic targets are skipped
acoustic.namespace†	/hertzino			hertzino_driver namespace
acoustic.duration_sec	5	s	1 .. 60	detection time when the target sets none
acoustic.margin_sec	20	s	1 .. 120	wait this long past the detection time
acoustic.status_max_age_sec	5	s	0.5 .. 60	status older than this: not connected
acoustic.corona	attention		normal, attention, warning, critical	severity of corona discharge
acoustic.surface	warning		normal, attention, warning, critical	severity of surface discharge
acoustic.floating	critical		normal, attention, warning, critical	severity of floating discharge
acoustic.intensity_warning	0		0 .. 1000	a discharge at least this intense is a warning; 0: off
acoustic.intensity_critical	0		0 .. 1000	a discharge at least this intense is critical; 0: off

A.13 Recorder

recorder_node

Parameter	Default	Unit	Range	Description
recordings_dir†	~/x30/recordings			a session per <YYYYMMDD>/<HHMMSS>_<reason>/
pose_topic†	/localization/pose			the robot's pose of each frame (map frame)
site_topic†	/site/status			the active site, in session.json
queue_size†	64		4 .. 10000	files waiting for the disk; more are dropped
thermal.enabled†	true			raw thermal frames (.npz) while recording
thermal.topic†	/hikmicro/thermal/temperature			32FC1, degC per pixel (hikmicro_driver)
thermal.ptz_topic†	/hikmicro/ptz/state			pan, tilt, zoom of each frame
thermal.min_interval_sec	0	s	0 .. 3600	at most a frame this often; 0: every frame

Parameter	Default	Unit	Range	Description
`video.enabled`†	`true`			the streams (.mp4) while recording
`video.host`†	`192.168.1.64`			the HIKMICRO camera
`video.port`†	554		1 .. 65535	its RTSP port
`video.user`†	`admin`			its user
`video.password_env`†	`HIKMICRO_PASSWORD`			environment variable with its password (no param)
`video.streams`†	`visible:101, thermal:201`			name:channel of rtsp://<host>/Streaming/Channels/<channel> (101, 201: main streams)
`video.urls`†	`[]`			more streams as name=url (any ffmpeg input), recorded as they are
`video.transport`†	`tcp`		tcp, udp	RTSP transport
`video.segment_sec`†	300	s	10 .. 3600	MP4 per segment, at a keyframe
`video.restart_sec`†	2	s	0.5 .. 60	a stopped stream restarts, up to 60 s
`video.ffmpeg`†	`ffmpeg`			the ffmpeg binary
`acoustic.enabled`†	`true`			the acoustic camera data while recording
`acoustic.namespace`†	`/hertzino`			hertzino_driver's topics
`acoustic.chunk_sec`†	10	s	1 .. 600	a .npz of each stream this often
`patrol.enabled`†	`true`			record while a route runs (AUTO, patrol)
`patrol.status_topic`†	`/navigation/status`			x30_navigation's status
`patrol.stop_after_sec`	30	s	0 .. 3600	no route this long: stop (laps stay one session)
`patrol.silence_sec`	60	s	1 .. 3600	no navigation status this long: no route runs
`disk.keep_days`	0		0 .. 3650	older sessions go; 0: kept while room is left
`disk.min_free_gb`	10	GB	0 .. 10000	the oldest sessions go while less is free
`disk.stop_free_gb`	3	GB	0 .. 10000	below this nothing more is written
`disk.check_sec`	30	s	1 .. 3600	disk checked this often while recording

A.14 Handheld gateway

`teleop_udp_node`

Parameter	Default	Unit	Range	Description
`port`†	9870		1 .. 65535	UDP port
`token_env`†	`X30_TELEOP_TOKEN`			environment variable of the shared secret
`robot_name`†	`""`			name in discovery beacons (“”: hostname)
`client_timeout_sec`†	2	s	0.2 .. 60	a silent handheld is dropped

Parameter	Default	Unit	Range	Description
max_vx†	0.8	m/s	0 .. 5	stick limit
max_vy†	0.4	m/s	0 .. 5	stick limit
max_wz†	1	rad/s	0 .. 10	stick limit
remote_source†	handheld			control mux source: control/remote/<source>/cmd_vel
telemetry_hz†	10	Hz	0.5 .. 50	state rate
robot_command_prefix†	/x30/			the driver's Trigger services (bringup's namespace)
robot_commands†	stand_up, sit_down, start_stepping, stop_stepping, gait_walk, gait_off_road, gait_stair, emergency_stop			the handheld's action menu, std_srvs/Trigger each
battery_topic†	/x30/battery			the driver's BatteryState
cmd_topic†	/x30/cmd_vel			the mux's command to the driver
driver_diagnostics_topic†	/x30/diagnostics			the driver's status line
health_diagnostics_topic†	/diagnostics			x30_health's findings (the computer's figures)
sites_dir†	~/x30/map			taught routes are listed from here
cameras†	visible=/hikmicro/visible/image_raw/compressed, thermal=/hikmicro/thermal/image_raw/compressed			name=topic of JPEG frames (CompressedImage); the first is shown
camera_fps†	5	Hz	0.5 .. 30	frame cap
camera_jpeg_quality†	60		10 .. 100	re-encoded frames
camera_max_width†	640		64 .. 4096	re-encoded frames
camera_max_bytes†	60000		1000 ..	bigger JPEGs are re-encoded
lidar_topic†	/cloud_registered_body			FAST-LIO's scan in body ('': no cloud)
lidar_hz†	2	Hz	0.1 .. 20	cloud rate
lidar_voxel†	0.1	m	0.01 .. 2	downsampling
lidar_max_points†	3000		10 .. 100000	points per cloud
navigation_node†	/path_follower_node			its speed/cruise is the patrol speed
patrol_speed_min†	0.1	m/s	0.01 .. 5	slowest patrol speed to set
patrol_speed_max†	0.6	m/s	0.01 .. 5	fastest (and robot limits/max_vx)

A.15 NETRA uplink

fleet_uplink_node

Parameter	Default	Unit	Range	Description
`server_url`†	`""`			NETRA, e.g. https://netra.example.com (“”: spool only)
`token_env`†	`X30_NETRA_TOKEN`			environment variable of the robot’s NETRA token
`robot_id`†	`x30-01`			fleet id of this robot in NETRA
`verify_tls`†	`true`			check the server certificate
`telemetry_hz`†	1	Hz	0.1 .. 10	snapshots
`upload_period_sec`†	5	s	0.5 .. 600	spooled uploads are tried this often
`spool_dir`†	`~/x30/spool`			routes and inspections wait here until NETRA has them
`sites_dir`†	`~/x30/map`			NETRA’s routes and plans go into the active site here
`clock_sync_period_sec`†	10	s	1 .. 3600	measure the offset to NETRA’s clock
`clock_warn_ms`	200	ms	0 ..	warn when the robot’s clock differs this much
`battery_topic`†	`/x30/battery`			the driver’s BatteryState
`health_diagnostics_topic`†	`/diagnostics`			x30_health’s findings (the computer’s figures)
`driver_diagnostics_topic`†	`/x30/diagnostics`			the driver’s status line

A.16 Simulator

`sim_robot_node`

Parameter	Default	Unit	Range	Description
`world_file`†	`""`			world YAML path or name in config/worlds
`scenario_file`†	`""`			scenario YAML path or name; empty: none
`driver_ns`†	`x30`			x30_driver namespace: /<ns>/cmd_vel, odom
`seed`†	0		0 ..	RNG seed (a scenario may set it)
`rate_hz`†	100	Hz	5 .. 1000	physics
`odom_rate_hz`†	50	Hz	1 .. 500	<driver_ns>/odom (leg odometry)
`publish_tf`†	`false`			odom -> base_link from the leg odometry
`max_dt`†	0.1	s	0.001 .. 1	longest physics step (stalls)
`cmd_timeout_sec`†	0.5	s	0.05 .. 10	older command: the body stops
`obstacle_radius`	0.25	m	0.01 .. 5	sim/obstacles size
`obstacle_height`	1.7	m	0.01 .. 5	sim/obstacles height
`lidar.rate_hz`†	10	Hz	0.5 .. 50	scans per second
`lidar.frame`†	`livox_frame`			frame of the merged cloud (= base_link)

Parameter	Default	Unit	Range	Description
`lidar.mounts`†	0.44367, 0, 0.05415, 0, 25, 0.43297, 0, -0.05519, 0, -46, -0.46576, 0, 0.03484, 180, -5, -0.41074, 0, -0.05601, 180, -56			per LiDAR x y z (m, base_link), dome azimuth and elevation (deg)
`lidar.azimuth_rays`†	360		8 .. 3600	rays per ring over 360 deg
`lidar.rings`†	16		1 .. 128	elevations over the vertical fov
`lidar.fov_min_deg`†	-7	°	-90 .. 90	lowest
`lidar.fov_max_deg`†	52	°	-90 .. 90	highest
`lidar.range_min`†	0.1	m	0 .. 10	closer: no return
`lidar.range_max`†	40	m	0.5 .. 500	no return
`lidar.noise_m`†	0.01	m	0 .. 1	range noise, 1 sigma
`lidar.point_dropout`†	0		0 .. 1	fraction of returns lost at random
`lidar.floor_returns`†	`true`			rays pointing down hit the floor
`lio.odom_frame`†	`camera_init`			FAST-LIO world frame
`lio.body_frame`†	`body`			FAST-LIO body frame (the IMU)
`lio.blind`†	0.7	m	0 .. 10	no points this close to base_link
`imu.rate_hz`†	200	Hz	0 .. 1000	0: off
`imu.topic`†	`/imu/data`			sensor_msgs/Imu, m/s^2 and rad/s
`imu.frame`†	`imu_link`			URDF frame of the IMU
`imu.xyz`†	0.13045, -0.001, 0.0149	m		imu_link in base_link (URDF)
`odometry.sigma_x`†	0.005	m/$\sqrt{\text{m}}$	0 .. 1	along-track noise
`odometry.sigma_y`†	0.005	m/$\sqrt{\text{m}}$	0 .. 1	cross-track noise (slip)
`odometry.sigma_yaw_deg`†	0.1	°/$\sqrt{\text{m}}$	0 .. 10	heading noise
`odometry.scale_error`†	0		-0.5 .. 0.5	measured / true distance - 1
`odometry.yaw_bias_deg_per_m`†	0	°/m	-10 .. 10	heading bias
`odometry.rotation_weight`†	0.2	m/rad	0 .. 10	turning counts as this travel
`leg_odometry.sigma_x`†	0.02	m/$\sqrt{\text{m}}$	0 .. 1	along-track noise
`leg_odometry.sigma_y`†	0.02	m/$\sqrt{\text{m}}$	0 .. 1	cross-track noise (slip)
`leg_odometry.sigma_yaw_deg`†	0.5	°/$\sqrt{\text{m}}$	0 .. 10	heading noise
`leg_odometry.scale_error`†	0		-0.5 .. 0.5	measured / true distance - 1
`leg_odometry.yaw_bias_deg_per_m`†	0	°/m	-10 .. 10	heading bias
`leg_odometry.rotation_weight`†	0.2	m/rad	0 .. 10	turning counts as this travel
`latency.lidar_sec`†	0.05	s	0 .. 1	cloud age
`latency.odometry_sec`†	0.01	s	0 .. 1	pose age
`latency.command_sec`†	0.02	s	0 .. 1	cmd_vel delay
`camera.rate_hz`†	10	Hz	0 .. 60	detections per second (0: off)
`camera.tags`†	`base_map`, `charging_dock`			world tags the detector publishes on TF

Parameter	Default	Unit	Range	Description
`camera.frame`†	`camera_color_optical_frame`			optical frame (x30_localization tag/)
`camera.publish_mount`†	`true`			base_link -> frame on /tf_static (false: the URDF has it)
`camera.mount_xyz`†	0.5, 0, 0.1	m		camera in base_link
`camera.yaw_deg`†	0	°	-180 .. 180	0: forward, 180: rear camera
`camera.fov_deg`†	80	°	1 .. 180	horizontal field of view
`camera.range_min`†	0.25	m	0 .. 10	closer the tag fills the image
`camera.range_max`†	4	m	0.1 .. 50	detection range
`camera.max_incidence_deg`†	70	°	1 .. 89	a plate seen more obliquely: lost
`camera.noise_m`†	0.003	m	0 .. 0.5	position noise
`camera.noise_deg`†	0.5	°	0 .. 20	yaw noise
`battery.start`†	0.9		0 .. 1	charge at start
`battery.drain_per_hour`†	0.2		0 .. 10	charge lost per hour
`battery.charge_per_hour`†	1		0 .. 100	charge gained per hour on the dock
`battery.report_charging`†	`false`			power_supply_status (x30_driver reports UNKNOWN)