– Confidential –

Qolo @LASA Hardware and Software Architecture

David Gonon, Vaibhav Gupta , Diego Paez G.
EPFL – STI – IMT – LASA

February 17, 2021

Abstract

This is a technical documentation and a platform to explore and plan future reactive collision avoidance developments for the semi-autonomous mobility device Qolo at LASA. It documents the basic design, which already contains the main software and hardware elements for reactive collision avoidance and pedestrian detection–tracking. It then describes further design options which integrate all the elements to enable real-time person perception and avoidance. Finally, it defines a development plan which encompasses a list of steps to take in order to complete particular functional requirements.

About

This document is part of the design and integration process for Qolo's control and perception systems for enabling real-time collision avoidance. To contribute, one requires access permission to the Overleaf project which hosts the central Git repository for developing this document. One can then clone it via the following command.

```
git clone https://git.overleaf.com/5ea2b05280d61a0001a1b7fb
```

Compiling the LaTeX code locally works by simply typing

```
make
```

given that all the used packages are installed. Assuming a `texlive` distribution, the following example shows how to use `tlmgr` to install a missing package.

```
tlmgr install hitec
```

Note that the package `hitec` [Billauer, 2001] provides the `hitec` document class giving this text its layout and format.

One can also include vector graphics in the document such as e.g. Fig. 1. The code underlying this example demonstrates how to import the image and text contents of *.svg files separately such that the latter are interpreted as LaTeX commands. It requires first to export the image and text contents separately from the *.svg file. The Makefile does this automatically for any *.svg file in the folder `svg/` when building the document with the following command.

```
make all
```

This process works for *.svg files saved by the software Inkscape. The only condition is that before saving all the text is selected and raised to the top layer.

Contributors shall add their names as authors.

Contents

List of Figures

1 Basic Design

The basic design refers to the current implementation on Qolo [Paez-Granados et al., 2018]. In comparison to the prospective final implementation, the basic design already contains most hardware and software elements (or their preliminary versions) without completely integrating them.

2 Sensor Architecture

The following describes the location of each sensor w.r.t. the robot's control coordinate frame at the center of the driving wheels (tf_qolo), which is at a height of Delta Z: 200.00mm.

Frontal FT sensor Location: X = 32.5 Y = 0 Z = -18.5

Front Lidar Delta X: 35.00mm, Delta Y: 0.25mm, Delta Z: 278.8mm Quaternion: [0, 0.0174524, 0, 0.9998477]

Rear Lidar: Delta X: -516.00mm, Delta Y: 0.00mm, Delta Z:163.7mm Quaternion: [0, 0, 1, 0]

Front Camera: Delta X: 116.46mm, Delta Y: 17.50mm, Delta Z: 241.04mm Angle: 83deg on y Quaternion: [-0.4685431 0.4685431 -0.5295917 0.5295917]

Rear Camera: Delta X: -588.22mm, Delta Y: 17.75mm, Delta Z: 108.76mm Angle: 78deg Quaternion: [-0.4592291, -0.4592291, 0.5376882, 0.5376882]

Front Left Camera: Delta X: -99.51mm, Delta Y: 67.57mm, Delta Z: 424.34mm Eurler Angles XYZ: [x: 0, y: 90, z: -90] Quaternion: [-0.5, 0.5, -0.5, 0.5]

Front Right Camera: Delta X: -99.45mm Delta Y: -37.63mm Delta Z: 424.42mm Eurler Angles XYZ: [x: 0, y: 90, z: -90] Quaternion: [-0.5, 0.5, -0.5, 0.5]

Fig. 1 shows the hardware and software elements which the basic design includes as well as their association to a specific computer.

Fig. 2 shows the coordinate frames which the basis design employs, their locations and orientations on Qolo (except for the external "world"-fixed frame), and which relative transformations are statically calibrated or dynamically estimated.

Fig. 3 shows the signal flows with spatial data contents and their processing by individual elements in the basic design. The schematic also indicates the coordinate frames which the signals' data contents refer to.

3 Composite Designs

This section explores different design options and provides details about the system components' functional properties that underlie these options.

3.1 Design Options

Consider the following options.

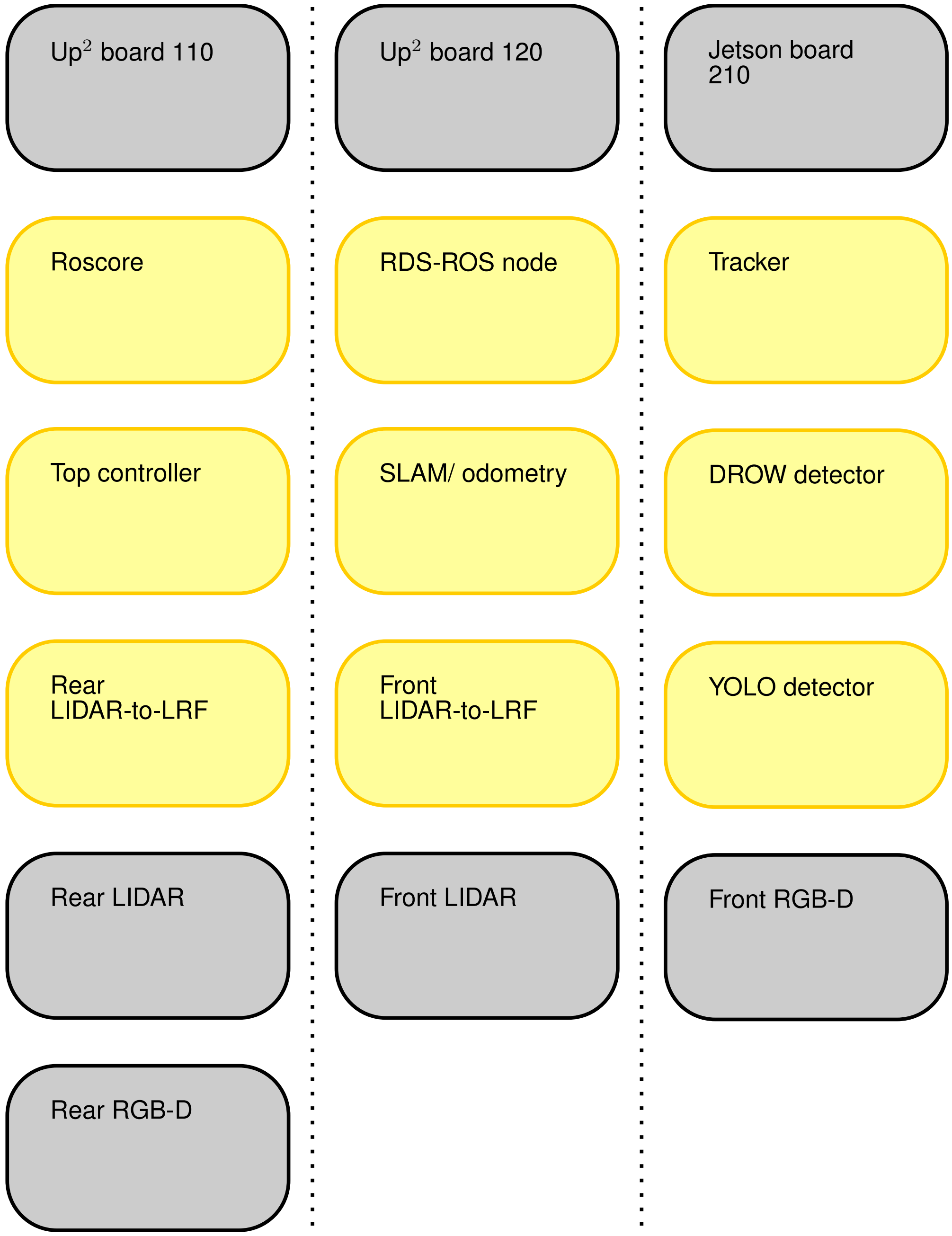

Figure 1: The basic design architecture, including hardware elements (grey) and software elements (yellow). Elements in the same column are associated to the same computer (top).

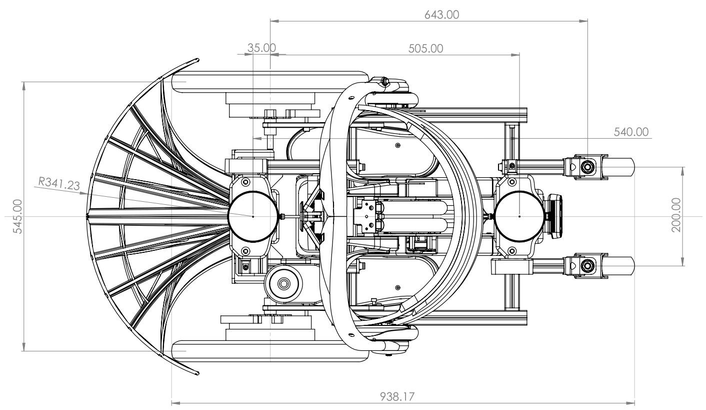

Figure 2: Coordinate frames for the basic design with statically calibrated () and dynamically estimated () relative transformations.

A Omni-directional LIDAR static-only perception, see Fig. 4. RDS treats each LIDAR measurement point as a static object.

B Front LIDAR pedestrian-only perception, see Fig. 5. RDS receives only pedestrian positions and velocity estimates from the tracker.

C Forward-facing pedestrian-only perception, see Fig. 6. This option is as the Option B just using also the RGB-D camera with the YOLO detector.

D Omni-directional pedestrian-only perception (under construction).

3.2 Components Properties

The following list describes particular components in more detail.

- *Tracker* Detects pedestrians and tracks their positions. In addition, it includes a velocity estimator for individual pedestrians to enhance collision avoidance.
- *Velodyne-laserscan* Selects one particular ring from a given 3D LIDAR measurement and converts it to a 2D LRF measurement.
- *RDS-ROS* Maps incoming nominal velocity commands to new velocity commands which avoid collisions with pedestrians and/or objects perceived by the tracker and/or other detection systems. The function which computes the new velocity command has the informal name “RDS–4” and is documented in the corresponding publication.
- *SLAM/odometry* Estimates and outputs the robot’s position and orientation with respect to a static frame which it chooses on-line and updates sporadically when

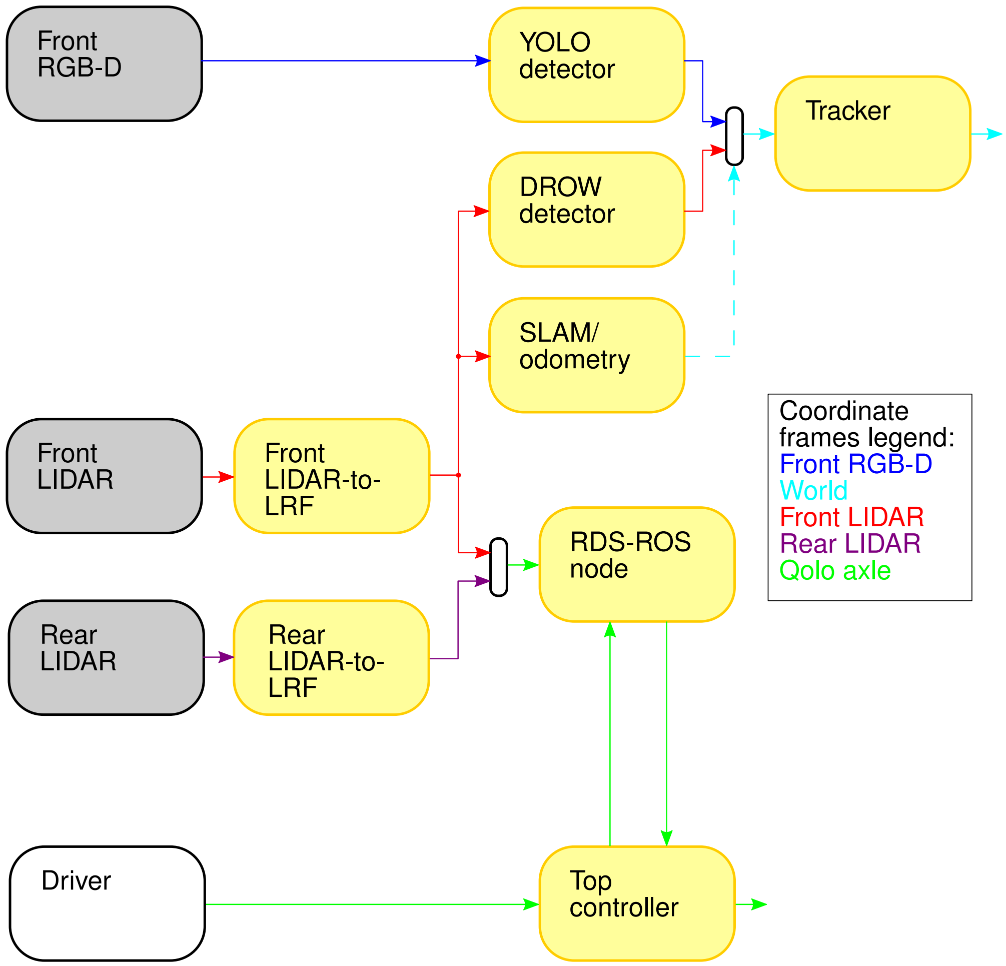

Figure 3: The basic design's spatial signal flows, their processing, acquisition, or generation by individual system components, and associated coordinate frames.

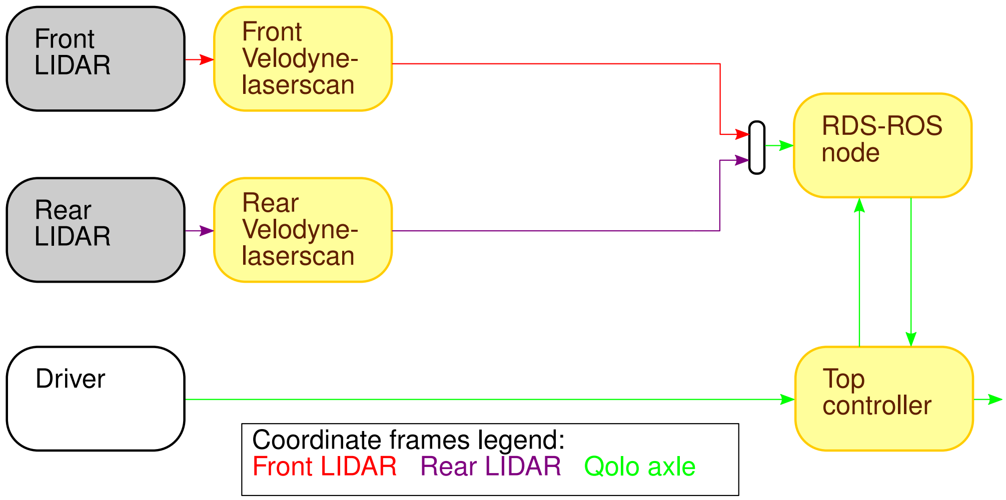

Figure 4: Spatial signal flows, processing, acquisition, or generation and associated coordinate frames for the composite design Option A.

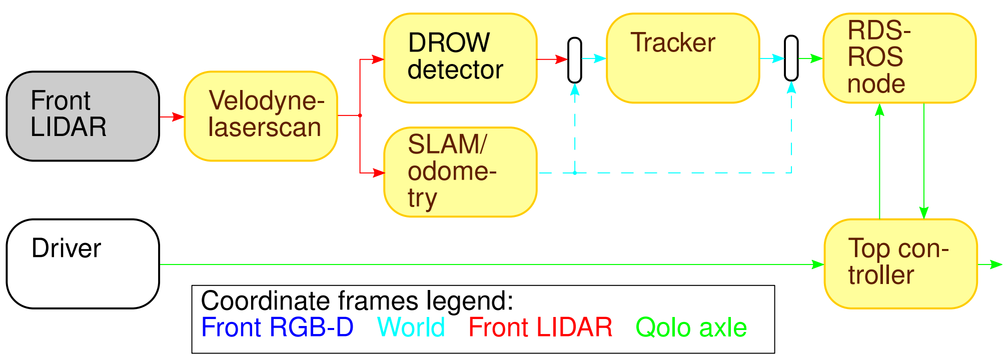

Figure 5: Spatial signal flows, processing, acquisition, or generation and associated coordinate frames for the composite design Option B.

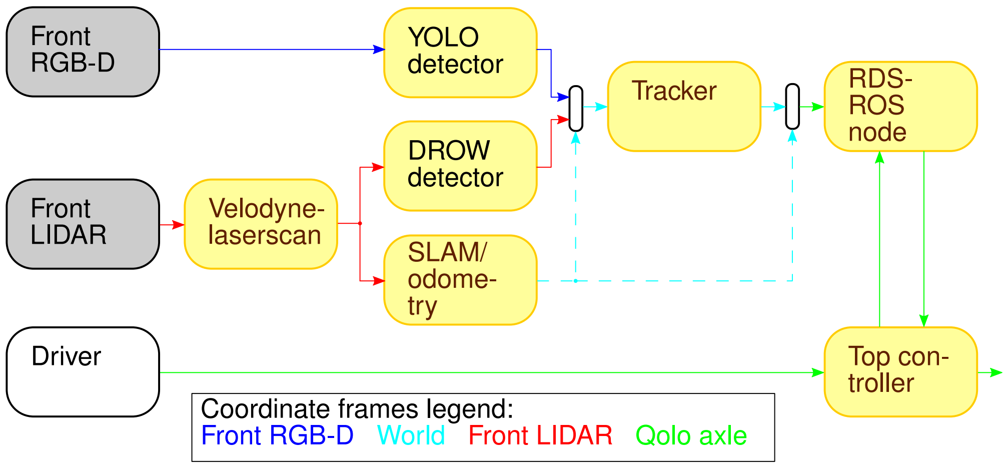

Figure 6: Spatial signal flows, processing, acquisition, or generation and associated coordinate frames for the composite design Option C.

necessary. It is necessary to transform spatial data to/from the static frame in which the tracker operates.

- *Top controller* Receives commands from the driving person using the embodied control [Chen et al., 2020] or remote joystick, forwards them to the collision avoidance system and transfers the response (the new command by the collision avoidance system) to the low-level controllers.

4 Development Plan

The functionality to develop and the plan to achieve it follow.

4.1 Functional Requirements

The system needs to perform the following functions.

1. In static environments, detect and avoid collisions with static objects.
2. In pedestrian environments, detect and track pedestrian positions, estimate their velocities, and on this basis, avoid colliding with them.

4.2 Pending Developments

The following developments are necessary to complete respectively for the above functional requirements.

1. Following either design Option B or C:
   (a) Pedestrian velocity estimator to append to the tracker.
   (b) RDS-ROS node interfacing with the tracker.

The following developments are necessary to complete in any case.

4.3 Tests

The following tests are necessary.

1. RDS-ROS input coordinate frame transformation correctness (for all design options).
2. Pedestrian velocity estimation accuracy.
3. RDS-ROS static objects avoidance (design Option A).
4. RDS-ROS static and dynamic pedestrian avoidance (design Option B or C).

5 Annex: Qolo's Dynamic Properties, Inertia Distribution

Qolo + Human Model 70Kg Mass: 120 Kg COM location: X = -210, Y = 0, Z = 585 Moments of inertia: (grams * square millimeters) Taken at the center of mass and aligned with the output coordinate system: Lxx = 49317302067.38 Lxy = -57564368.51 Lxz = -2664151405.57 Lyx = -57564368.51 Lyy = 44040518777.15 Lyz = -74821976.08 Lzx = -2664151405.57 Lzy = -74821976.08 Lzz = 30981109380.82

Qolo without Driver Mass: 53 Kg COM location: X = -150, Y = 0, Z = 180 Moments of inertia: (grams * square millimeters) Taken at the center of mass and aligned with the output coordinate system: Lxx = 22330096364.79 Lxy = -41094938.09 Lxz = -77685727.35 Lyx = -41094938.09 Lyy = 17810790368.50 Lyz = -23241317.20 Lzx = -77685727.35 Lzy = -23241317.20 Lzz = 29030064794.02

Qolo's Main Frame Mass = 48.5 Kg COM location: X = -160 Y = 0 Z = 184 Moments of inertia: (grams * square millimeters) Taken at the center of mass and aligned with the output coordinate system: Lxx = 22237227707.13 Lxy = -40068359.11 Lxz = 23000350.44 Lyx = -40068359.11 Lyy = 17580113808.49 Lyz = -23871393.07 Lzx = 23000350.44 Lzy = -23871393.07 Lzz = 28829812595.28

Wheels Mass = 1.5 Kg / each

Front Bumper Inertia Mass = 1.5 Kg Center of mass: (millimeters) X = 0.21 Y = 168.82 Z = 39.40 Moments of inertia: (grams * square millimeters) Taken at the center of mass and aligned with the output coordinate system. Lxx = 21550778.48 Lxy = 8599.93 Lxz = 3328.12 Lyx = 8599.93 Lyy = 33325434.16 Lyz = 5754224.13 Lzx = 3328.12 Lzy = 5754224.13 Lzz = 42160389.50

References

[Billauer, 2001] Billauer, E. (2001). hitec – Class for documentation. https://ctan.org/pkg/hitec. ↑ p. i

[Chen et al., 2020] Chen, Y., Paez-Granados, D., Kadone, H., and Suzuki, K. (2020). Control Interface for Hands-free Navigation of Standing Mobility Vehicles based on Upper-Body Natural Movements. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2020)*, Las Vegas, USA. IEEE Xplorer. ↑ p. 6

[Paez-Granados et al., 2018] Paez-Granados, D. F., Kadone, H., and Suzuki, K. (2018). Unpowered Lower-Body Exoskeleton with Torso Lifting Mechanism for Supporting Sit-to-Stand Transitions. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2755–2761, Madrid. IEEE Xplorer. ↑ p. 1