

Chapter 1

Program Structure

The program is divided in several nodes in order to keep the extensive program structured and clear. Between the nodes topics with a different data structure are used to pass on processed data.

1.

Node The laser range finder provides a tool to make a 2D map of the environment.

Topic The laser data is published with a rate of 40Hz and uses the LaserScan message type. Each array sent is 1080 measurements of type float64 long. The laser range finder scans in CCW direction so the first measurement is located at -45° , which is actually the body of pico.

Laser scan data = $[1 \ 2 \ \dots \ 1080]$

2.

Node This node receives the rough laserdata and then filters and smoothes this. Smoothing can be done by a moving average filter using a Gaussian or Rectangular window. The filtering means that for example data points that are further away than 3 meters are canceled out.

Topic The processed data is published with a rate of 40Hz. It uses the LaserScan message type but the length of the array is dependent on the size of the moving average window.

Filtered laser scan data = $[1 \ 2 \ \dots \ (1080 - \text{window width})]$

3.

Node This node calculates the local and global maxima and minima from the laser data.

Topic The maxima and minima are published with a rate of 40Hz. The message type is specifically designed for this topic and exists of 4 vectors containing the minima and maxima and the angles at which they occur. The length of these vectors depends on the environment of the robot.

Values Maxima = $[Value1 \ Value2 \ \dots \ ValueN]$

Indices Maximum = $[Index1 \ Index2 \ \dots \ IndexN]$

Values minima = $[Value1 \ Value2 \ \dots \ ValueN]$

Indices minima = $[Index1 \ Index2 \ \dots \ IndexN]$

4.

Node The number and order of the maxima and minima depends on the environment of the robot. This node analyses the data and determines on what type of crossing the robot is situated. This node tells the next node which turning possibilities there are; left, right and/or straight.

Topic Each possibility, left, right or straight, gets it's own boolean. When all three are true the robot is situated at a crossing, when only two are true the robot is situated at a T-crossing and when none of them are true the robot is situated in a dead end. The publishrate of this topic is 40Hz.

boolean left (1.1)

boolean right (1.2)

boolean straight (1.3)

5.

Node This node is kind of the brain of the program. This node makes all decisions about the turns. In the end the 'map making' will be done in this node. This node tells the robot to turn or to drive. While all corners are made by turning on the base the robot has never angular and linear velocity.

Topic Turn This topic tells the base to turn left of right. The datatype is an integer, 0 for no turn, 1 for left turn, 2 for right turn.

Topic Drive Safe This topic tells the base to turn left of right. The datatype is an boolean, 0 for no forward velocity, 1 for forward velocity.

6.

Node This node takes care of the corners.

Topic This topic tells the base to drive or not. The datatype is the Twist vector message type.

7.

Node

Topic This topic tells the base to drive or not. The datatype is the Twist vector message type.

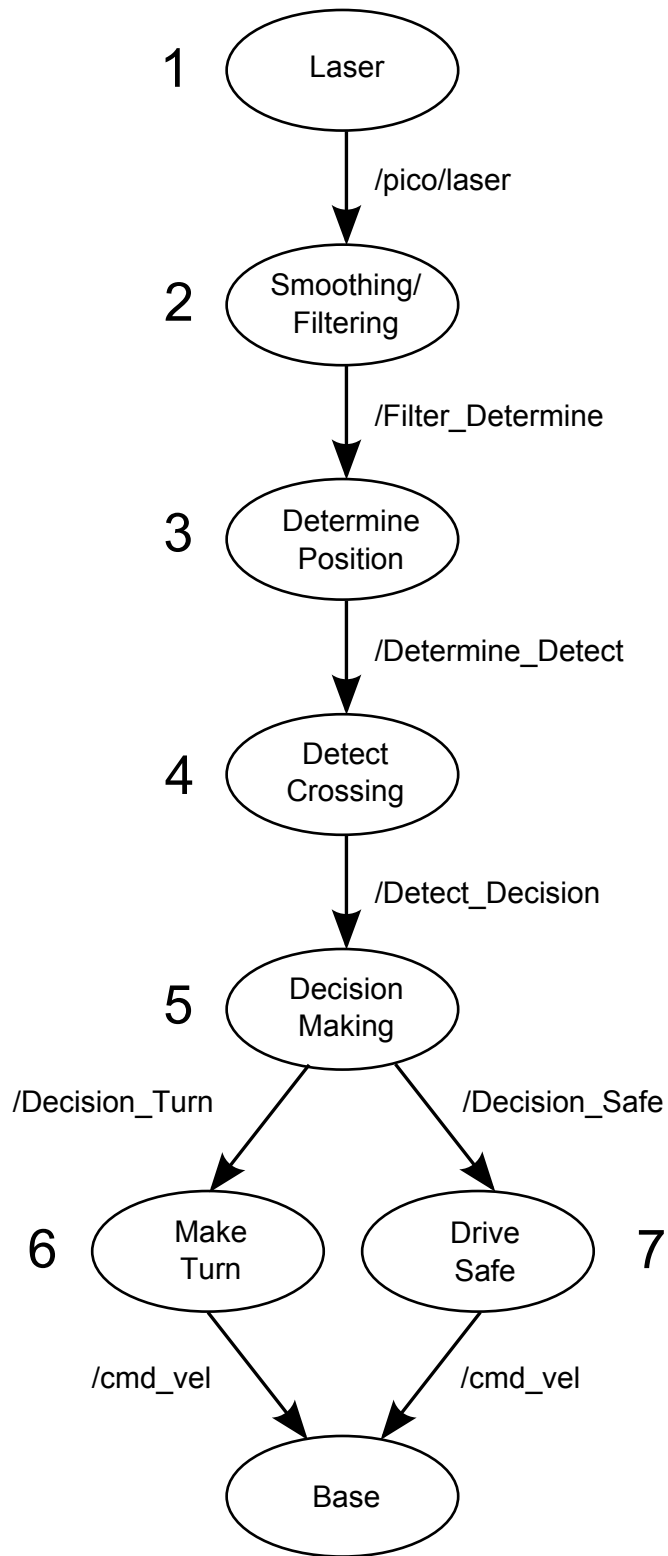


Figure 1.1: Program divided into nodes.