



Sewer **I**nspection **a**utonomous **r**obot

D28.13 – SIAR Robotic Solution Description

SIAR Consortium

IDMind (IDM), PT
Universidad de Sevilla (USE), ES
Universidad Pablo de Olavide (UPO), ES



Table of Contents

1. Introduction	3
2. SIAR Robot Platform V4	4
2.1. Robot Features and Components	4
2.1.1. Locomotion Platform Main Features	4
2.1.2. Sensors	5
2.1.3. Actuators	8
2.2. Platform Design and Mechanics	9
2.2.1. Traction system and suspension	10
2.2.2. Width Adjustment Mechanism	11
2.2.3. Cameras' Positioning and Brackets	12
2.2.4. Robotic Arm with HD camera	12
2.2.5. Main body protective enclosure	13
2.2.6. Anti-falling optional mechanism	16
2.3. Robot Architecture	17
2.3.1. Electronic Power Architecture	18
2.3.2. Communication Architecture	20
2.4. Locomotion Platform Electronics	21
2.4.1. SIAR Motor control and Linear Motor boards	21
2.4.2. Sensor&Management and Battery control boards	23
3. Communications	25
3.1 Manually Deployable Repeater	25
4. Software Architecture	28
4.1. SIAR Control Station	29
4.2. Details on the proposed communication middleware	32
4.3. Localization module	33
4.4. Navigation module	35

4.4.1. Local Navigation	35
4.4.2. Navigation module in the presence of forks	36
4.5. Sewer Map Building	39
4.5.1. Local maps	40
4.5.2. Global 3D maps	41
4.6. Map Analysis	42
5. Report on the final demo of December, 13th	46
5.1 Inspection plan for Passeig de Garcia Fària	46
5.2 Localization results	47
5.3 Navigation	48
5.4. Communication experiments	49
5.4.1 Experiments at Passeig de Garcia Fària, December 2018	49
5.5. Inspection report	50
5.5.1 Serviceability Defect 1 (Alerts 0 and 1)	51
5.5.2 Structural Defect 1 (Alert 2)	52
5.5.3 Structural Element 1: fork (Alert 3)	52
5.5.4 Structural Defect 2: inlet with sediments (Alert 4)	53
References	55

1. Introduction

The SIAR robot platform V4 (version 4) was concluded at the end of September 2018. Its first public exhibition was at IROS 2018 (see Figure 1.1) and the first row of experiments with this new platform were performed during the third week of October. These experiments were then followed by the experiments related with the structural inspection evaluation (7th of November). Finally, in the 13th of December 2018, was performed the demo related with the final evaluation of Phase III of the project.

In this last version of the hardware platform, special account has been taken on the improvement of the robustness of mechanical connections, the choice of materials and the insulation of the sensitive parts from the ingress of dust and liquids. Aesthetic considerations were also taken in consideration.



Figure 1.1 - SIAR V4 at IROS 2018 (Photo: Evan Ackerman/IEEE Spectrum)

The document is as organized as follows. Section 2 presents the SIAR robot platform V4, its main features and components, mechanical design, electronics, low-level communication architecture and robot safety. Section 3 details the communication system that endows the SIAR robot with continuous wireless connection with the base station. Section 4 presents the software architecture and makes a description of the methodology used in each software module. Section 5 presents the experimental report on the final demo of Phase III.

2. SIAR Robot Platform V4

SIAR V4 is a six-wheeled robot platform with the ability to change the width between its wheels to address the different sections of the sewer network. The robot is fully equipped to move autonomously for about 4 hours while executing the inspection operation. The following section describes the hardware configuration of the solution.

2.1. Robot Features and Components

This subsection introduces the main features and devices included in the SIAR robot.

2.1.1. Locomotion Platform Main Features

SIAR robot platform main specifications are the following:

- Robot Kinematics: 6 wheeled with 6 independent locomotion motors and 1 linear motor for width adjustment.
- Weight (with batteries): 65 Kg
- Battery autonomy: >4 hours
- Maximum Velocity: 0.75 m/s
- Acceleration: 1 m/s²
- Emergency Stop Acceleration: 3,3 m/s²
- Dimensions with maximum width (Height x Max_Width x Length) : 44 x 85 x 88 cm
- Dimensions with minimum width (Height x Min_Width x Length) : 44 x 52 x 1080 cm

The SIAR robot system has an onboard computer and an external Control Center computer:

- The Internal Computer (aka NUC) is the main onboard processor that gets the information from the onboard sensors and automatically generates the proper commands for safe navigating through the environment. Moreover it tracks the approximate location of the robot based on the methodology described in Section 5. It communicates with the base station through a 2.4GHz link. This communication includes sensing and status information to the main station, processing data from the encoders and receiving the commands from the operator from the Control Center Computer.
- The Control Center Computer (aka Control Center) consists in a regular Laptop computer with an additional power source, to provide increased autonomy, and a communication systems, that will allow real-time communication with the robot at any time during the mission. It provides relevant data about robot monitoring, receiving system alerts and perform automatic sewer monitoring and analysis in real time.

Table 2.1 lists the devices that are present in the SIAR solution.

Robot processing	Quantity
NUC i7 computer (AKA NUC)	1
External Robot Wireless Link	1

Sensor and Management system	Quantity
Main Board	1
Switch Board	1
Battery	4
DCDC converters	3

Motor Actuation	Quantity
Main board + 6 motor drivers	1
Add in width motor board + 1 motor driver	1
Traction motors + encoders	6
Width Motor + potentiometer	1

Sensors and Actuators	Quantity
IMU sensor	1
RGBD Camera	7
Lights Projectors	4
Robotic arm	1
Arm HD camera	1
Arm high-bright LEDs ring	1

Control Station	Quantity
Control station computer	1
External Control Station Wireless Link	1

Wireless repeaters deploy system	Quantity
Wireless repeaters	3

Table 2.1. SIAR devices.

2.1.2. Sensors

The SIAR robot is equipped with perception, navigation and gas sensors. For navigation the robot uses encoders to control the velocity of the traction motors and a potentiometer to control the position and velocity of the platform width actuator. An inertial sensor gives the orientation of the robot. The depth cameras are able to give an estimation of the obstacles/walls distances to the robot, and are also used to refine the odometry estimates. For perception the robot uses the RGBD cameras to analyze the sewer

environment and detect any abnormal situation and also a HD camera mounted in the top of a small robotic arm.

Navigation Sensors

The robot is able to safely navigate in the environment by using the measures provided by different sensors. The robot makes use of RGBD cameras, encoders odometry and the IMU sensor to estimate its position and orientation. Obtaining a good estimation of the robot's displacement and orientation is crucial not only when generating a 3D map of the environment, but also when performing complex maneuvers in such a challenging scenario as the sewers are. The RGBD cameras can be used also for obstacle avoidance.

List of navigation sensors:

- Inertial sensor IMU: Arduimu v3 with the team IMU filtering
Function: Orientation estimation
Position on Robot Platform: in the robot's center of rotation
- Motor encoders: 500 pulses encoder
Function: Odometry calculation
Position on the robot: one installed in each motor
- 4 x RGBD cameras: 4 Orbbec's Astra S with 6 m range
Function: Reactive navigation and obstacle detection
Position on Robot Platform: two side-by-side cameras on the front tilted 45° down and two side-by-side cameras on the rear tilted 45° down.
- 2 x RGBD cameras: 2 Orbbec's Astra with 8m range
Function: Operator awareness and visual odometry
Position on Robot Platform: one on the front tilted 5-10° up and one on rear tilted 5-10° up.

Perception Sensors

The robot uses the RGBD cameras for sewer monitoring and analysis. The perception sensors are the following (please note that the first six cameras are also used for navigation purposes):

- 4 x RGBD cameras: 4 Orbbec's Astra S with 6 m range
Function: Obstacle detection and space geometry analysis
Position on Robot Platform: two side-by-side cameras on the front tilted 45° down and two side-by-side cameras on the rear tilted 45° down.
- 2 x RGBD cameras: 2 Orbbec's Astra with 8 m range
Function: 3D reconstruction
Position on Robot Platform: one on the front tilted 5-10° up and one on rear tilted 5-10° up.

- 1 x RGBD camera: 1 Orbbec's Astra with 8 m range
Function: Manhole localization and 3D ceiling reconstruction
Position on Robot Platform: Top of the robot pointing to the ceiling.
- 1 x Camera: HD camera
Function: Inspection and help on robot maneuvers
Position on Robot Platform: Robotic arm installed on top of the robot.

Environmental Sensors

A Waspnote Pro OEM gas system was installed on the robot to measure the gases concentrations inside the sewers. Figure 2.2 shows the acquired Waspnote Pro system.

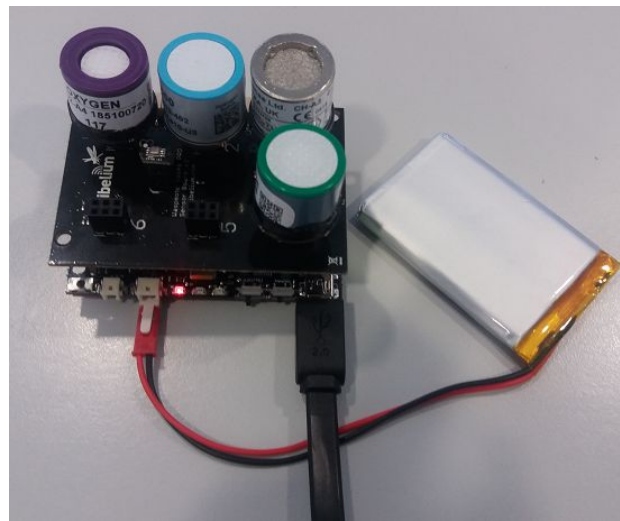


Figure 2.2 - Waspnote Pro OEM gas system

The sensor can measure the following gases:

- Molecular Oxygen (O₂) Gas Sensor [Calibrated]. It will provide the percentage of oxygen on the environment with an accuracy as good as $\pm 0.1\%$ (ideal conditions). It will give an alarm if the percentage drops under 19.5% or if it rises above 23.5%.
- Carbon Monoxide (CO) Gas Sensor for High concentrations [Calibrated]. It will provide the ppm of the CO concentration with an accuracy as good as ± 1 ppm (ideal conditions). It will give an alarm if the ppm are above 50 ppm.
- Hydrogen Sulfide (H₂S) Gas Sensors [Calibrated]. It will provide the ppm of H₂S concentration with an accuracy as good as ± 0.1 ppm (ideal conditions). It will give an alarm if the ppm are above 5ppm.
- Methane (CH₄) and Combustible gases sensor [Calibrated]. It will provide a measure of LEL methane percentage with an accuracy as good as $\pm 0.15\%$ LEL (ideal conditions). It will give an alarm if the percentage is above 50%.

The system also includes a temperature ($^{\circ}\text{C}$), humidity (% RH), pressure sensor (Pa) that will be used to increase the accuracy of the gas sensors. Figure 2.3 shows a datalogin from the sensors. In addition to the gas sensors information previously described, it is also possible to determine the inclination of the board, using an 3-axis accelerometer, the voltage and power level of the battery.

```

+++++
Acceleration X: -16 | X angle: -0.9167715072
Acceleration Y: -87 | Y angle: -4.9910416603
Acceleration Z: 1000 | Z angle: 90.0000000000
Battery Level: 85 % | Battery (Volts): 4.0840001106 V
Gas O2 concentration: 20.7667961120 Vol. % - O2 Alarm: OFF
Gas H2S concentration: 0.5267459869 ppm - H2S Alarm: OFF
Gas CO concentration: 0.4259382247 ppm - CO Alarm: OFF
Gas CH4 concentration: 0.0000000000 % LEL - CH4 Alarm: OFF
Temperature: 25.8799991607 Celsius degrees
RH: 47.8994140625 %
Pressure: 99022.8593750000 Pa
*****

```

Figure 2.3 -SIAR gas sensors datalogin example

2.1.3. Actuators

The robot is equipped with actuators for locomotion, for the visual inspection around the robot and to actuate the width adjustment mechanism.

Locomotion Actuators

For locomotion the platform uses six motors to drive the 6 wheels and one linear motor to change the width of the robot.

- 6 x Maxon RE 35 90W 15V motor with a Maxon GP 32 HP 28:1 Gearbox and encoder HEDS 5540 with 500 pulses
 - Function: providing independent locomotion to each wheel of the robot
- One Linear motor Linak LA23 with internal potentiometer and limit switches.
 - Function: change the width between wheels and the centre of mass of the robot.

Inspection arm with HD camera

The inspection of specific areas around the robot is complemented and improved with the use of a HD camera mounted over a 5 DOF robotic arm. This arm is located on the top of the robot. Figure 2.4 shows SIAR using the arm to inspect the walls on the left side of it.



Figure 2.4 - Using the inspection camera arm

Illumination LED light projectors

Due to the poor illumination conditions in the sewer, the SIAR robot is equipped with LED light projectors. The operator's camera requires a minimum of illumination which is provided by four 6W LED projectors (two in the front and two in the rear of the robot). The camera of the robotic arm is also equipped with a ring of nine high-bright LEDs. Figure 2.4 shows the use of the front LED projectors and the ring of LEDs from the arm.

2.2. Platform Design and Mechanics

The SIAR robot is a six-wheeled differential robot with a width adjustment mechanism. A central structural frame is used to connect the motorized wheel traction system, the width adjustment mechanism, the batteries and to carry the payload equipment and electronics. The wheel traction system is composed of 6 sets of independent motors, 90° worm gearboxes and 260 mm offroad wheels. Each of these sets has a independent suspension arm that connects to the central robot frame. The width adjustment mechanism, actuated by a linear motor, increases the adaptability of the robot to the sewer configuration. The seven

cameras described above are connected to the frame with the use of the camera link supports which maintain the cameras at the correct height and inclinations.

A fiberglass upper body was introduced to protect the electronic components from the dirty waters and wastes. This body provides some protection to the cameras. The robotic arm with the HD camera is installed on the top of the fiberglass body. Figure 2.5 depicts the final SIAR solution.



Figure 2.5: SIAR robot.

2.2.1. Traction system and suspension

The ground clearance near the wheels of SIAR V4 has been increased while maintaining the same suspension stroke. The traction system uses a motor connected to a worm gear box connected directly to the robot wheels. The bottom of the worm gear plastic cover is rounded to facilitate the slippage of the robot on the forks' gutter edges. The connection between the worm gear box and the arms uses linear bearings mounted on the worm gear box and stainless steel rods with suspension springs are mounted on the wheel arms (Figure 2.6).



Figure 2.6 - New wheel and arms connection.

2.2.2. Width Adjustment Mechanism

The width adjustment mechanism improves the robot adaptability to different structural changes in the working environment. This mechanism allows to vary the width of the robot by means of a change in the position of the wheels. This mechanism is actuated with the use of a linear motor. The linear motor is installed and protected inside the central frame. The wheels' arms actuation is done also in the same level. This operation increases the central clearance to the floor while protecting these sensible parts. Figures 2.7 and 2.8 show the width adjustment system, with the minimum width (left) and the maximum width (right).

The linear motor is linked to a linear rail. The variation of the stroke of the motor makes the carriage move along the rail and in this way changing the width of the robot.

With this mechanism the width of the robot can vary from 520 mm (minimum) to 850 mm (maximum). The width variation implies a length variation of the platform. For the minimum width of 520 mm the robot has its maximum length of 1080 mm. For the maximum width of 850 mm the robot has its minimum length of 880mm. The height of the robot is independent of the the width of the robot.



Figure 2.7: width adjustment system.



Figure 2.8: minimum width 520 mm (left); maximum width 850 mm (right).

The mechanism described above proved to be very important and effective in the sewer environment. It allows the robot to adjust its width in order to pass through sections of different width and at the same time change the center of gravity of the robot. This feature is very helpful when crossing bifurcations, allowing the robot to maintain the balance and also to overcome large gaps in the sewer. The displacement of the center of the frame in relation with the two center wheels has an amplitude of 40 cm (-20 cm to +20 cm).

2.2.3. Cameras' Positioning and Brackets

The following cameras and configurations are used in SIAR:

1. Four Astra S cameras (two front and two rear) tilted 45 degrees downwards for gathering information of the floor in the surroundings of the SIAR platform. Two cameras are needed due to their small field-of-view (FoV). By using the two cameras is possible to have a combined 110° FoV in each direction (backwards and forwards). Resulting FoV is slightly lower than the sum of two cameras, to avoid blind areas. The information of these cameras is used for reactive navigation. In this way the SIAR has real-time knowledge of the position of the gutter and the obstacles. This information is critical to guide the robot safely across the sewers. Also they can be of great use to detect and precisely localize serviceability losses in the sewer system.
2. Two Astra cameras for operator awareness and for 3D reconstruction. These two cameras (front and rear) are placed horizontally and can provide long-range information (up to 8 m) for the localization and navigation algorithms. The horizontal camera is tilted up slightly (5-10 degrees) to minimize the interferences between cameras.
3. One Astra camera pointing to the ceiling. This camera is used to reliably detect the manholes of the sewers, improving the performance of the localization system of the SIAR platform. The camera can be also used to extract 3D information of the ceiling when required.

Plastic brackets maintain the camera assembly in the correct position. The brackets grant the correct horizontal and vertical positioning of the cameras (see Figure 2.9). The central camera, pointing to the ceiling also has a bracket, although, this one is fixed without angular regulation.



Figure 2.9: Back and Frontal cameras set.

2.2.4. Robotic Arm with HD camera

The robotic arm carries a HD camera surrounded by a ring of 9 high-bright LEDs. This set, that is fixed on the top of the robot, is used for the inspection of areas that are not accessible by the other cameras on the robot. The robotic arm has five degrees of freedom. The arm is composed of two segments. The HD

camera is attached to the end of the second segment over a pan&tilt head. Figure 2.10 depicts the robot using the camera mounted on the robotic arm for a closer view of the sewer wall. The illumination of that area is provided by the ring of LEDs mounted around the HD camera.



Figure 2.10: SIAR's inspection robotic arm with HD camera

A driver has been written to control the arm at a high level of ROS. The arm driver can read or write the arm joint position, read the torque or the temperature of each motor, or make sure the robot does not try to move to an unreachable configuration. Furthermore, functions are provided to move the robot either between two points along the fastest path or in a straight line. There are also functions that allow to move the arm to another pose or move it to another joint position. To do this, the robot's inverse and forward kinematics have been calculated, making use of its dynamic equations.

2.2.5. Main body protective enclosure

The final robot design includes a protective fiberglass shell to enclose and isolate all the electronic components, sensors, batteries, lights and motors. Figure 2.11 shows the electronic components inside SIAR, that are enclosed by the new shell.

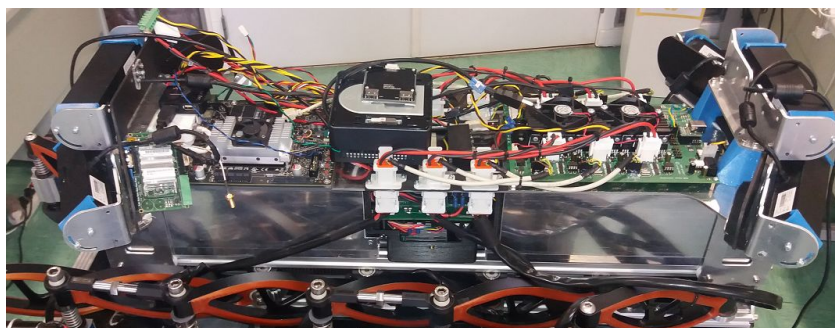


Figure 2.11: SIAR without the protective shell

The top RGBD camera is located on the top under a protective transparent polycarbonate and the other 6m RGBD cameras have been individually isolated against the ingress of water or other liquids. All the

RGBD cameras are involved by the shell to prevent contact with the walls. Figure 2.12 shows the robot with the protective shell ready to be deployed.



Figure 2.12 - SIAR with the protective shell

On the top, two aluminium bars were added to facilitate the deployment/retrieval of the robot through the manhole, preventing the top cover from rubbing against the walls and steps while the robot is deployed through the manhole access (see Figure 2.13). The same bars will protect the inspection arm with camera from being damaged while the robot passes through the manhole. The arm motors and camera are protected against the ingress of water.



Figure 2.13 - Siar robot descending to sewer using the aluminum bars to protect the components

SIAR is powered by two batteries packs (Figure 2.14), one to power the motors and another to power the electronics. Each pack contains two 12V 15Ah LifePO4 batteries. The batteries are installed inside the batteries packs and have been isolated against the ingress of water.

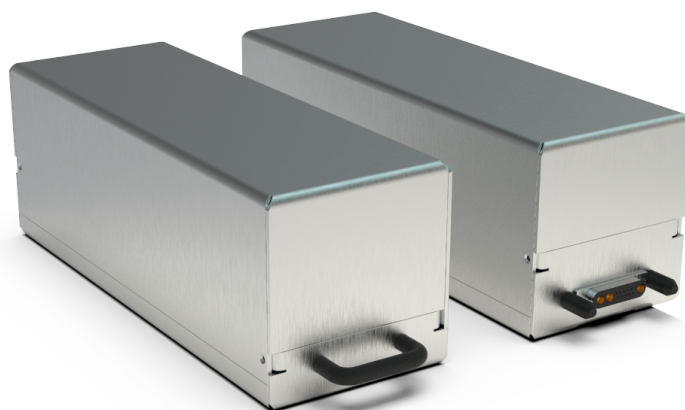


Figure 2.14 - SIAR battery pack

The packs are installed in drawers as depicted in Figure 2.15 (one in the front and another in the back). The pack enters the drawer in a sliding motion and performs an isolated contact with the robot electronics. This mechanism provides the robot with an easy and fast way to swap the batteries during a longer inspection routine.

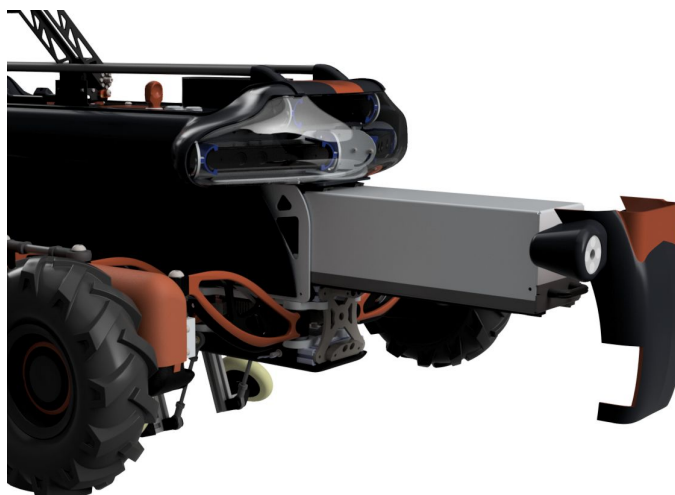


Figure 2.15 - Battery pack drawer

The motors' power and sensors connect with the main isolated body through IP67 power and signal connectors (see Figure 2.16). The illumination LED projectors are installed inside IP66 enclosures.

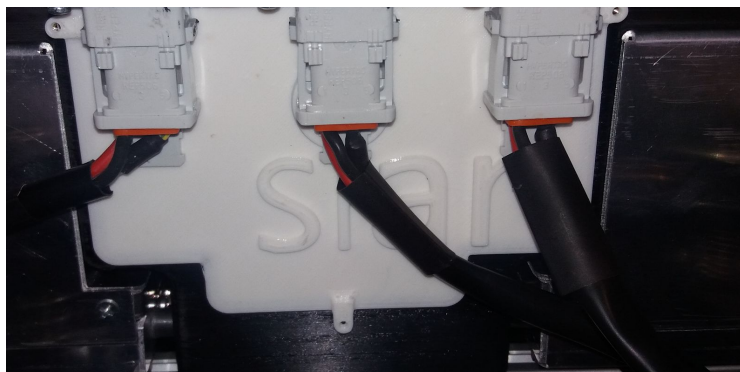


Figure 2.16 - IP67 power and signal connectors.

To ease the deployment/collection of the robot, one eye bolt is fixed to the robot structure, as shown in Figure 2.17.



Figure 2.17 - Using the eye bolt to descend the SIAR robot

2.2.6. Anti-falling optional mechanism

An optional mechanism has been studied to prevent the robot from falling inside the gutter. The mechanism is composed of two pairs of wheels that can be mounted on the robot's front and rear (see Figure 2.18). This mechanism can use the lateral sides of the gutter to mechanically limit the deviation of the robot from the center of the gutter. In the forks, these wheels will force the robot to go up, adjusting the position of the robot to the center of the gutter. The mechanism is passive and should be manually adjusted to the type of gutter where it will be used. This mechanism should only be used in sewer galleries with a middle gutter.

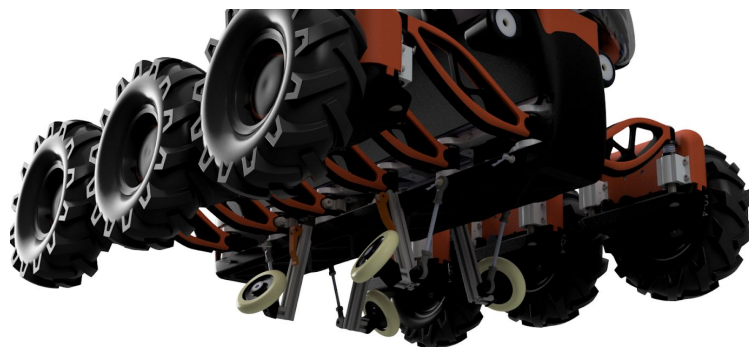


Figure 2.18 - Anti-falling optional mechanism

2.3. Robot Architecture

The system is composed by the following components:

- **NUC i7.** i7 based computer responsible for all the onboard processing. It can communicate with the remote Control Center through a 2.4GHz link to send sensor information and receive user commands.
- **Four Astra S cameras and three Astra cameras.** These cameras are the main sensors for navigation, localization and structural analysis.
- **SIAR Motor control board.** Connected to the six motors of the platform and sends the encoder data to the NUC i7. It is connected to the NUC i7 through the USB hub. Described in Section 2.4.1.
- **SIAR Linear Motor board.** Is an extension board that connects to the SIAR Motor control board and allows the control of the width of the robot. Described in Section 2.4.1.
- **SIAR Sensor&Management board.** Connects to NUC i7 through a USB hub. This board is used to control the power inside the robot, control the robotic arm, control the robot lights and connect and process the environment sensors. Described in Section 2.4.2.
- **SIAR Battery Control board.** Connects to the SIAR Sensor Management board being responsible for the connection between the batteries and electronic and motor system.
- **IMU.** An ardu-IMU v3 is used as IMU in order to estimate the 3D-orientation of the platform in real-time. This allows to control the platform taking into account dynamic and kinematic effects. It also improves the localization of the platform. It is connected through the USB -hub.
- **Microhard Nvip 2400 WiFi link.** This link offers a high bit rate (up to 45Mbps), long-range (up to 10 Km with Line of Sight (LoS)) and robust WiFi communication link. This link will provide several RGB and depth video-streams to the Base station as well as the odometric information. It is connected to the NUC through an Ethernet connection (RJ-45).

The following subsections describe the power and communication architectures between all the electronic device components.

2.3.1. Electronic Power Architecture

The robot platform is powered by 2 packs of two LiFePO4 12V 15AH batteries. One pack of batteries delivers power to the motor drivers and the other provides energy to all other electronics (e.g. computer, electronics and sensors). Each battery from the pack, depicted in Figure 2.19, has the following specifications:

- Voltage: 12 V (12.8 V nominal)
- Capacity: 15 AH
- Charging Voltage: 14.6 V
- Charging Current: < 3 A
- Max Continuous Discharging Amperage: 15 A
- Maximum Discharging Current: 30 A
- Lifecycle of single cell: 2000 cycles.
- Low voltage (minimal): 9.2V
- Includes an embedded battery management system (BMS) circuit.



Figure 2.19. 12V 15AH LiFePO4 battery pack

There is a total of approximately 0,36 KWh of installed power in each pack of batteries that provides an autonomy of 4 hours of operation. The pack of batteries are installed in a way that can be easily accessed and replaced by another charged pack to continue with the operation.

The batteries can be charged inside the robot (slow charge) or outside (fast charge). A power supply and charger can be connected to the robot, allowing to set up the robot, program or get data from it while charging the batteries packs. In this case a 5A LiFePO4 charger is used to charge both sets in 12 hours. Another option is to remove the battery packs and connect each pack directly to a 5A LiFePO4 charger. Each pack will charge in 6 hours.

The batteries and the power in the robot are managed by the Sensor&Management Board that measures the battery levels, battery charge, and also controls the units (motors, sensors and actuators) powered by the batteries. Figure 2.20 depicts the onboard power architecture.

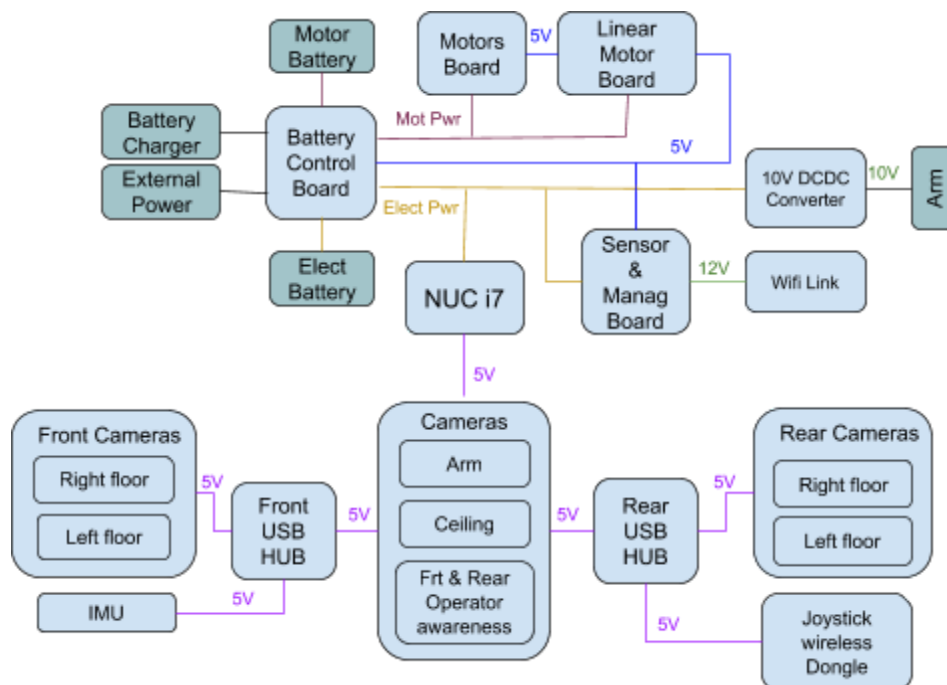


Figure 2.20. Onboard Power Architecture

Inside the NUC a voltage converter transforms the electronic power voltage to 5 volts. The 5V are used to power all the cameras, IMU and joystick dongle. The Sensor&Management board has a 5V and a 12V DC/DC power regulator that will be used to power the electronics boards. An additional power regulator is used to supply 10 V to the Robotic arm motor controllers.

Tables 2.2 and 2.3 show the equipment that is connected to batteries and their maximum current and power consumption.

Battery	Powered item	Nominal Current (Ah)	Power (Wh)
Motor (2,8Ah)			
12V 30Ah	Traction wheels motors	2,5	30
360W	Linear width change motor	0,3	3,6
Total		2,8	33,6

Table 2.2: SIAR motor battery and maximum power usage

Battery	Powered item	Nominal Current (Ah)	Power (Wh)
Electronics (2,8Ah)	NUC i7	2,33	28
12V 30Ah	Sensor&Management Board	0,2	2,4
360W	Motor Board	0,15	1,8
	7x Orbbec Astra Camera	2,6	18,2
	Inertial Sensor	0,05	0,25
	2x Led Light Projectors (each side)	1	12
	Robotic Arm	0,05	0,6
	Wireless link	0,15	1
Total		6,53	64,25

Table 2.3: SIAR Electronic battery and maximum power usage

From the table is possible to understand that it is expected a higher current consumption in the electronic part. With all the system running at the maximum, the electronic battery will go from fully charge to empty in approximately 4 to 5 hours.

Energy consumption

Each of the SIAR batteries is monitored using the actual voltage and the overtime integrated current. With this information and using the discharge curves for the LiFePO4 batteries, it is possible to estimate the remaining time of operation for each battery. Because the robot uses two packs of batteries, the remaining operation time is given by the set that has a lower value. The Sensor&Management board calculates all the data for each battery set.

2.3.2. Communication Architecture

On the SIAR robot the center of the Hardware Architecture is a NUC i7 Computer that processes all the information given by the cameras and navigation sensors and sends actuation commands to Motor board and Sensor&Management Boards. This computer has 3 independent USB controllers, that are able to control 2 USB ports each. This means that the computer has 6 USB ports. Each USB port allows to control 2 cameras without losing the image quality. As can be seen in the Electronics' architecture, 4 cameras are connected directly to computer and 2 USB HUBs connect the other 4 cameras. The HUBs are also used to connect the existing boards and components. A RJ-45 connection is used to connect to the WiFi link that communicates with the Control Center. Figure 2.21 shows the adopted communication architecture.

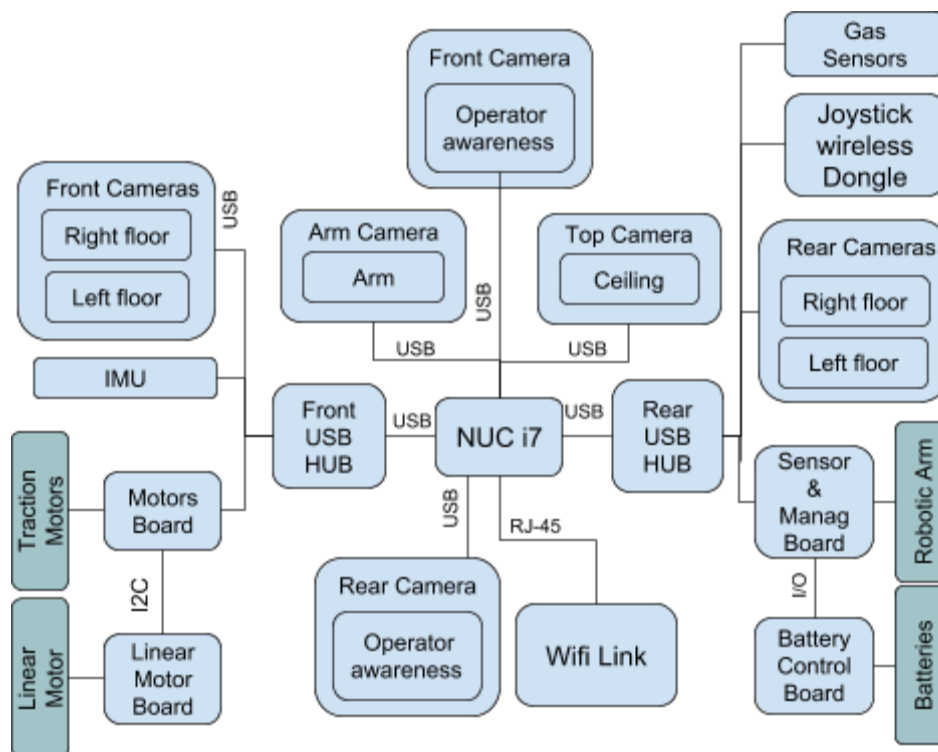


Figure 2.21: Communication architecture

2.4. Locomotion Platform Electronics

Several electronic boards were designed for SIAR which are described in the following subsections. SIAR platform includes the following electronic boards:

- SIAR Motor control board;
- SIAR Linear Motor board;
- SIAR Sensor&Management board;
- SIAR Battery Control board.

2.4.1. SIAR Motor control and Linear Motor boards

The Motor Controller Board manages the robot locomotion. The Master Motor Controller uses a microcontroller to control and manage all the communication between the NUC and the Slave Motor Driver Controllers. There are seven slave motor control drivers: six that control the locomotion wheels and one that controls the width of the robot. The Slave Controllers Motor Driver Controllers use a microcontroller to provide all the necessary control signals to the motor drivers. The overall architecture of the Motor Controller Board is depicted in Figure 2.22. The Linear Motor Control board is an extension Slave board of the SIAR Motor Board, able to control the Linear Motor that changes the robot width. The Motor Power is used to power the motor drivers and the 5 V from the Sensor&Management board is used to power all the electronics.

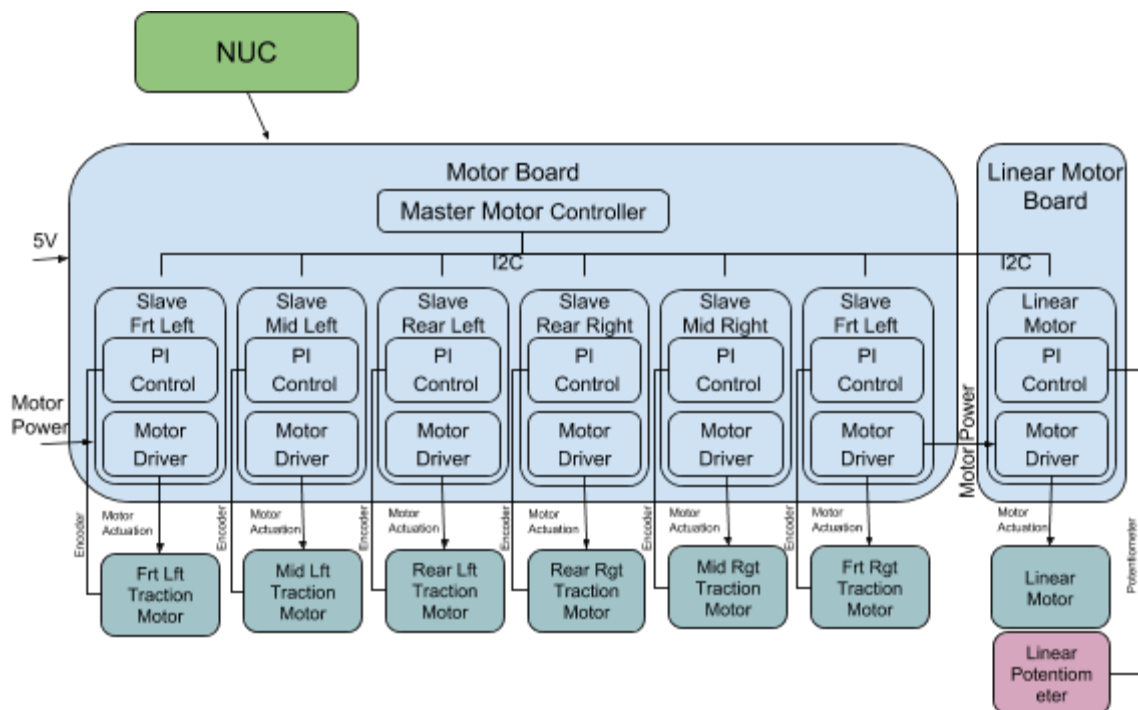


Figure 2.22: Motor Board architecture

Master Motor Controller tasks:

- runs low-level control loops to check for critical changes in the motor system that can affect the robot operation;
- provides an I2C bus Observer that checks the information received from the I2C Slave devices to understand faults in the communication and/or on the devices;
- checks the proper operation of each Slave Controller through exchange of status information;
- monitors the drivers' power supply;
- exchanges information with the NUC;
- enables/disables the motor drivers;
- enables/disables the QuickStop control of the motor drivers.

Slaves PI Traction Motor Controllers tasks:

- receive velocity references from the Master Motor Controller;
- periodically read the encoder pulses;
- send the read encoder pulses to the Master Motor Controller;
- calculate the error between the reference velocity and the read velocity;
- implement a PI controller to calculate the motor actuation;
- implement an anti-windup error limit;
- implement a reference velocity acceleration/deceleration profile;

- remove the dead zone of the motor;
- limit the maximum velocity of the motors.

Slave PI Platform Width Motor Controller tasks:

- receives width and velocity references from the Master Motor Controller;
- periodically reads linear potentiometer value;
- sends actual width to the Master Motor Controller;
- calculates the error between the reference velocity and the read velocity;
- implements a PI controller to calculate the motor actuation;
- implements an anti-windup error limit;
- implements a reference position and velocity acceleration/deceleration profile;
- removes the dead zone of the motor;
- limits the maximum velocity of the motors.

The Master Motor Controller receives velocity and width commands from the computer and returns the encoder pulses and the width values. The Controller connects to traction and width PI microcontrollers that generate the control actuation to follow velocity and width references. Each traction microcontroller connects to the motors using a power H-bridge, and provides the pulses measured by encoders. The width microcontroller connects to the motors using a power H-bridge, and provides the width measured by a potentiometer.

Each microcontroller is optically isolated from the motor driver using a high-speed optocoupler for the control actuation signals and an optical amplifier for the current measurements. It is also optically isolated from the computer communication port, again using a high-speed bidirectional optocoupler. Several low-level fault diagnostics have been implemented to detect problems in the normal working of each component or lack of communication.

2.4.2. Sensor&Management and Battery control boards

The Sensor&Management Board is responsible for the power management and sensor acquisition. It receives orders from the onboard robot NUC and returns information about the batteries, sensors and actuators. The Sensor&Management Board Architecture is depicted in Fig. 2.23.

The Sensor&Management Controller uses a microcontroller to manage all the communication with the high-level robot NUC and control all the actuators and sensors devices connected to the Sensor&Management Board. This includes the robotic arm and lights.

The Sensor&Management Board is responsible for managing all the power system, including:

- measuring the energy level in each set of batteries;
- managing the connection to the external charger;
- controlling the charge of each battery.

The Sensor&Management Board controller is able to communicate with the NUC using a USB-to-RS232 converter.

The Sensor&Management Controller will gather information from the sensors and will run low-level control loops that will check for critical changes in the environment or system that can affect the robot operation.

The Battery Control Board is a Sensor&Management slave board. This board samples the consumed current and voltage from each battery. This information is supplied to the Sensor&Management board to be used for battery level calculation. It also provides the needed electronics to switch from battery power to external power and charge.

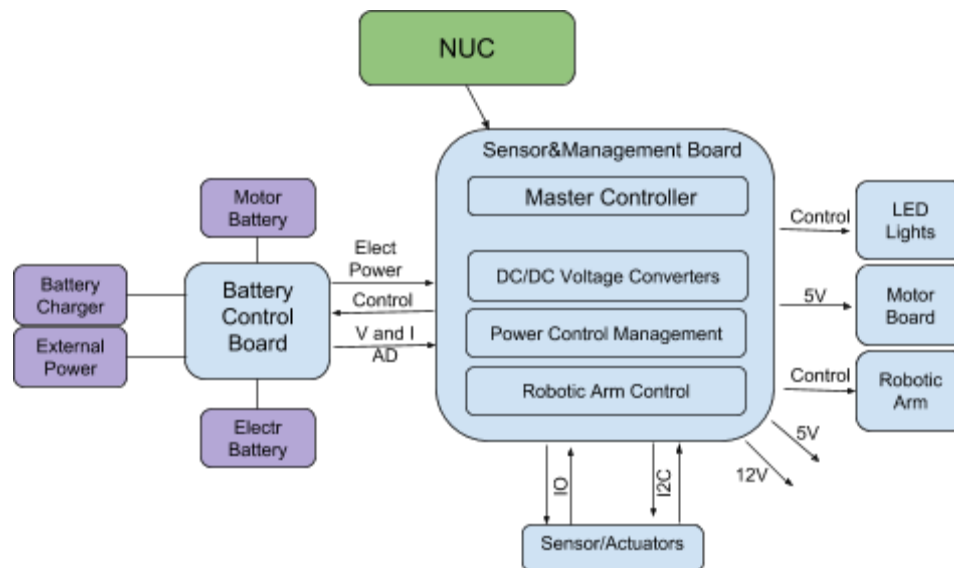


Figure 2.23. Sensor&Management Board architecture.

3. Communications

The SIAR inspection makes use of manual deployment of repeaters to increase the working range of the solution. The manual deployment of the repeaters is quick and precise, and it can be performed by a single worker. The following section describes the design of the deployable repeater.

3.1 Manually Deployable Repeater

The design allows a quick deployment of the repeaters without the need to go inside the manhole to deploy and fix them inside the sewer. Figure 3.1 shows the new wireless repeater design.



Figure 3.1 - Wireless repeater for manual deployment.

The operator that carries the repeater just has to open the manhole cover, adjust the repeaters height to the floor and fix it to the manhole first step or to the manhole cover. It takes less than 2 minutes to deploy and retrieve the repeaters. In this way the traffic impact is reduced. Figure 3.2 on left shows the deployment of the repeater on the manhole steps, on the right shows the same repeater installed on the manhole cover.

As depicted in Figure 3.1 the system is composed of two parts:

- Top part - includes the battery, the ON/OFF button, the step fixing system and the positioning/power cable reel;
- Bottom part - carries the wireless repeater and a sensor to determine if the repeater is at its optimum position.

The repeater system uses a 11.1V 2650 mA battery that permits an operational time of near 5h. This battery is installed on a battery compartment installed on the top part of the repeater. Table 3.1 shows the repeater's battery specifications.

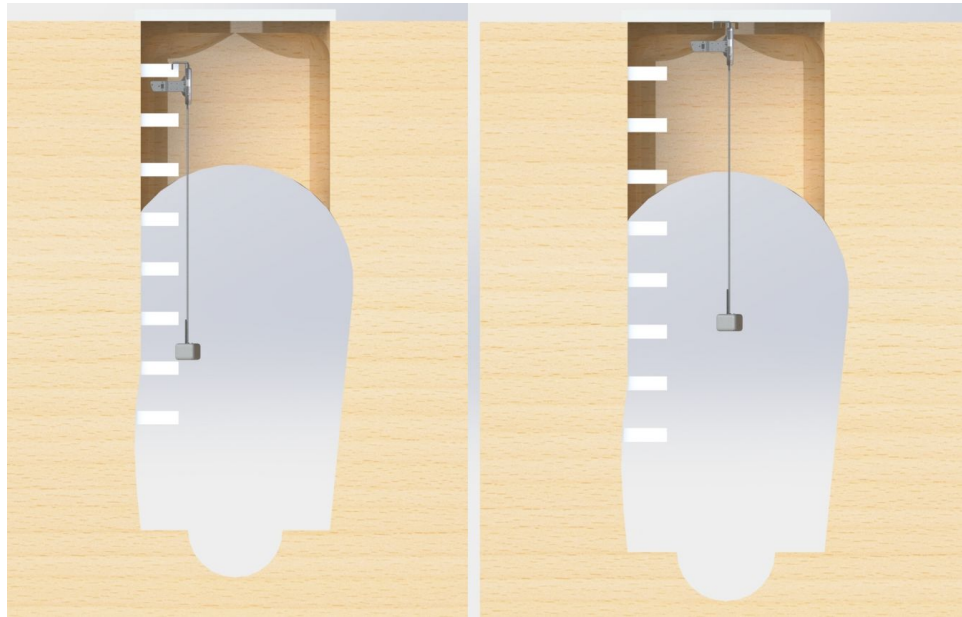


Figure 3.2 - Left: Installation of the repeater on the steps of the manhole.
Right: Installation of the repeater on the manhole cover;

Battery Description Manufacture - Type	Capacity (mA)	Oper. Time (h)	Dimensions (H x L x W mm)	Weight (g)
Turnigy - 3S 11.1V Lipo Transmitter pack	2650	4h50	105x29x26	148

Table 3.1 - Repeater's battery specification

The bottom part, shown on Figure 3.3, includes a distance sensor and the wireless repeater.



Figure 3.3 - Repeater bottom part

The distance sensor is an unidirectional micro LIDAR with a detection range of 12m. This sensor is installed on the bottom of the repeater box, measuring the distance to the sewer floor (not the gutter). The optimum position will be given by a small circuit board that will use the measured distance and activate 3 different LEDs. The blue LED indicates that the system should be raised, the red LED indicates that the system should be lowered and the green is activated when the system is in the right position.

Due to the fact that the wireless repeater consumes a max of 0.5A and the distance sensor consumes 0.13A with peaks of 0.8A, they do not work at the same time. The Top switch will have two working positions:

- Position "I" - powers the repeater power supply and connects the red LED, near the switch;
- Position "II" - powers the distance sensor and connects the green LED, near the switch.

Figure 3.4 shows the repeater top part, which has a shape that allows the system to be hanged on the steps, by adjusting the two bars to the step size and position, and two magnets on the top to alternatively fix the system to the manhole metallic cover.

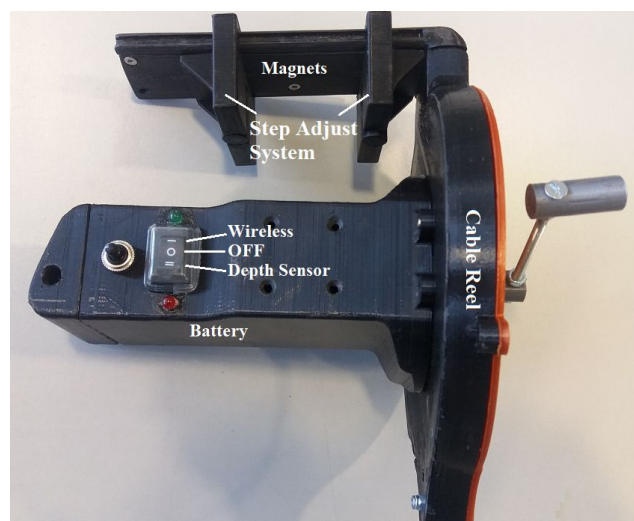


Figure 3.4 - Repeater top part

The performance of the system has been tested in the Final Demo performed in the 13th of December, 2018. Please refer to Section 5.3 for the analysis of the results. Furthermore, in phase II of the project we carried out experiments to validate the communications using several NVIP2400 repeaters in an Ad-Hoc network mounted in the Mercat del Born scenario. In this case we managed to cover all the inspection area by deploying the base station communication system and three additional repeaters. Please refer to deliverable D28.6, Section 3.3, for an in-depth analysis of the results. In addition, we present further results of the performance of the system in the Final Demo of December, 13th in Section 5.4.

4. Software Architecture

The SIAR software architecture is a set of modules that make use of the robot platform, sensors and communications, described in the previous sections, to develop the required functionalities (self-localization, navigation, inspections, etc). This architecture is implemented under the Robotic Operating System¹ (ROS) framework, using the ros-kinetic distribution under Ubuntu 16.04.

The software modules reflect the functionalities required for the sewer inspection, as established in [ECHORD++, 2014]. Figure 4.1 shows the main modules of the architecture of the final system.

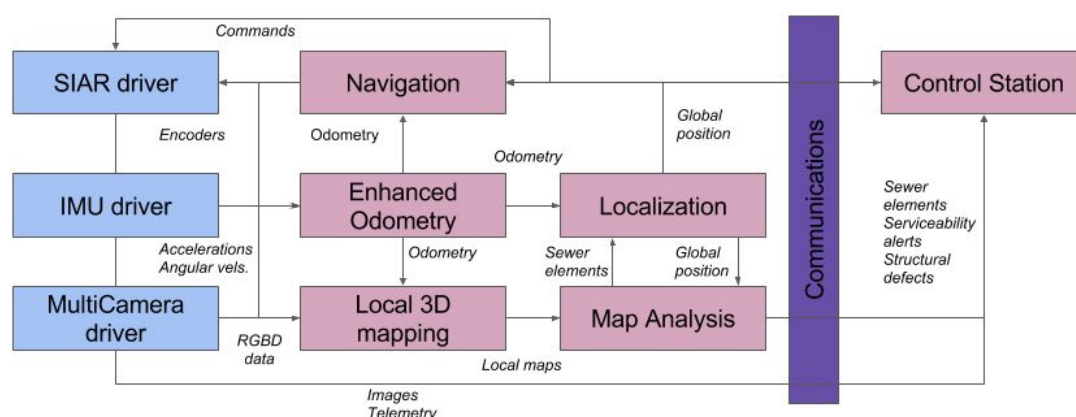


Figure 4.1. Software Architecture

These modules are the following:

- SIAR driver. Software that interfaces with the hardware board of the robot described in Section 2, including the arm, and offers standard ROS commands and services for easy integration of the robot with the rest of the system.
- IMU driver. This package filters the raw data produced by the inexpensive IMU sensor Arduimu v3.
- Multi-camera driver. This package extends the functionality of the `openni2_camera`² and `openni2_launch`³ ROS packages. It provides the user with an utility for generating a launch file that sequentially activates the onboard RGB-D cameras. Each camera can be configured with its own setup, i.e. changing the image rate, its resolution and many other parameters.
- Enhanced Odometry. This package provides enhanced odometry estimates by fusing IMU and encoders measures with visual odometry estimates. In this way, the SIAR platform is able to

¹ <http://ros.org>

² http://wiki.ros.org/openni2_camera

³ http://wiki.ros.org/openni2_launch

precisely and robustly estimate a 6-DOF odometry. A good odometry estimation is necessary for obtaining precise 3D maps of the environment.

- Control Station (Section 4.1): This module allows the operator to control the robot (in manual and semi-autonomous mode) and receive all the relevant information, including real-time video feeds, 3D data and alerts when critical conditions occur. These conditions can include automatic detection of structural defects, bad state of the radio links, low battery indicator, the robot has lost traction and should be recovered, among others. It also displays the localization of the robot within a map of the zone.
- Communication block (Section 4.2): even though ROS offers a middleware for inter-process communication, we experienced that it was not very optimized for real-time transmissions where the delay should be as low as possible. Therefore, to design and implement new ROS modules that allow real-time communication within the robot and the base station with minimum delay.
- Localization (Section 4.3). The precise odometry estimations are used by this module in order to estimate the global position of the platform in geographical coordinates. This block uses prior information of the sewer systems (e.g., localization of manholes, inlets, etc.), and the detection of such elements provided by the map analysis module, in order to correct the drift which is accumulated in the odometry estimation and to provide the location of the robot within the sewer network.
- Navigation (Section 4.4). This module is in charge of the autonomous navigation modes of the SIAR robot. It employs the sensory input to estimate non-transversible configurations and automatically avoid them. It includes planning capabilities to overcome obstacles and determine maneuvers. This module also allows the operator to command actions to the SIAR arm.
- Local 3D mapping (Section 4.5). This package gets the enhanced odometry outputs and uses them to integrate the measurements from the different RGB-D sensors disposed over the SIAR robot. This would allow the operator to have a precise 3D local reconstructions of the sewer. With the localization outputs, these local maps can be referred to the global sewer network.
- Map Analysis (Section 4.6). This module provides the high level information required for sewer inspection. It receives the local 3D maps and analyzes them to extract sewer elements, estimate the serviceability and inspect critical defects.

4.1. SIAR Control Station

During Phase III we developed and experimentally tested the final teleoperation GUI. This GUI has been designed according to the specifications obtained during the visit of the 12th of April, 2018 and during the Serviceability demonstration. Currently the Base Station is able to provide to the user, in real time, information related with the Serviceability and Structural Defect Analysis module (see Section 4.6). Moreover, the operator has the option of hiding or showing some outputs of the module as desired. Finally, the operator has now the information of the three front cameras available and the inspection camera which has been requested to confirm the alerts obtained by the Serviceability Analysis module.

The final version of the GUI includes the following features:

- All the functionalities are now compiled into a standalone application which has been designed to be user friendly.
- Three different modes of operation: exploration, inspection and mission execution & planning.
 - The exploration mode has a new design that allows the operator to have localization, real-time images, depth information and SIAR prioceptive information at the same time (see Figure 4.2). This mode is recommended to take metric measurements with the RGB-D data.
 - A new inspection mode has been included (see Figure 4.3), which also includes the new inspection camera installed in the arm (see Section 2.2.5). This mode is recommended for navigation and for obtaining details of possible defects.
 - In the mission execution mode the operator can easily get information about the alerts found during the experiment (see Figure 4.4).
- In addition, the content of the GUI is customizable by the user and new modes can be saved to further adapt to the needs of each operator.

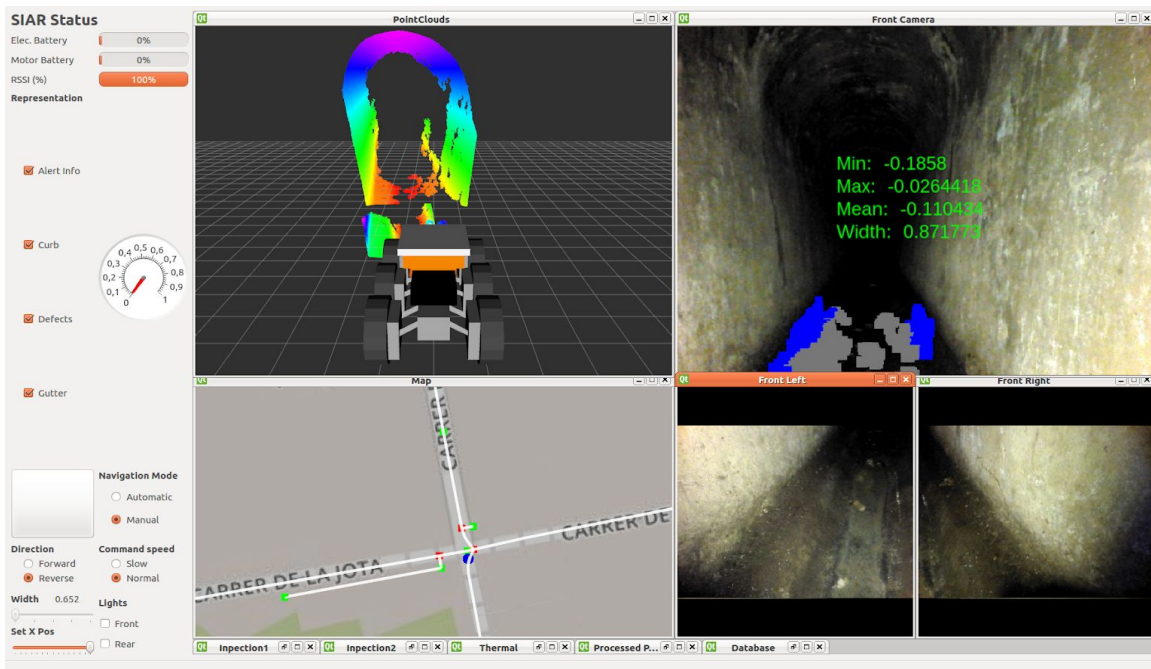


Figure 4.2: Exploration view of the control station.

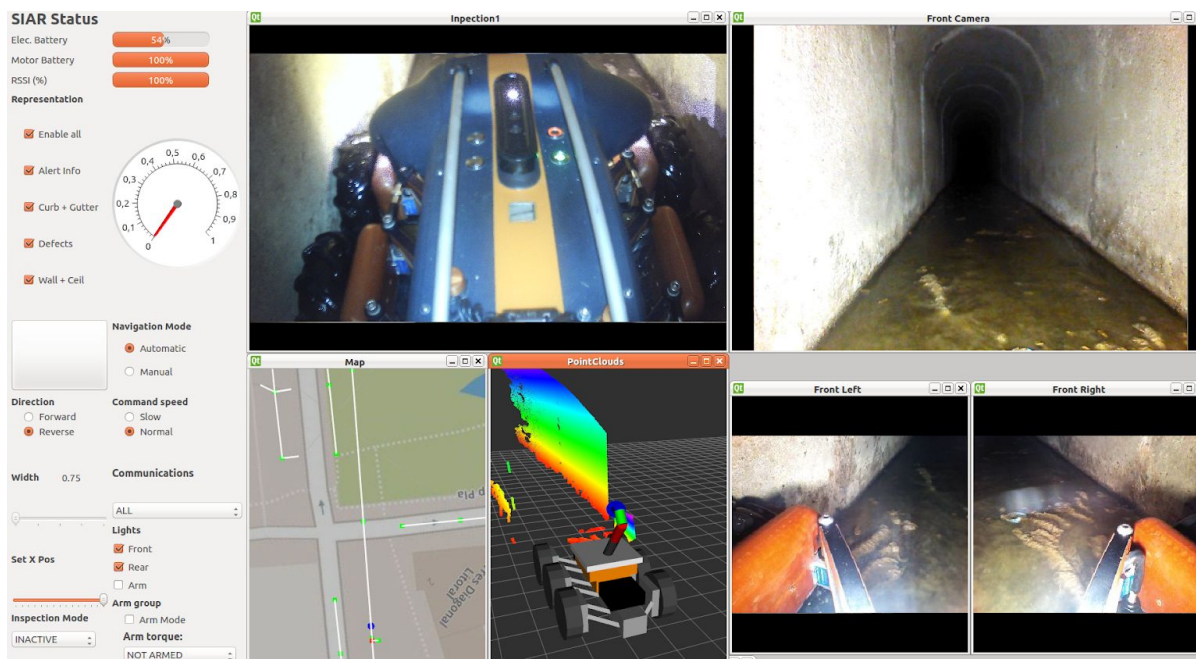


Figure 4.3: Inspection view of the control station.

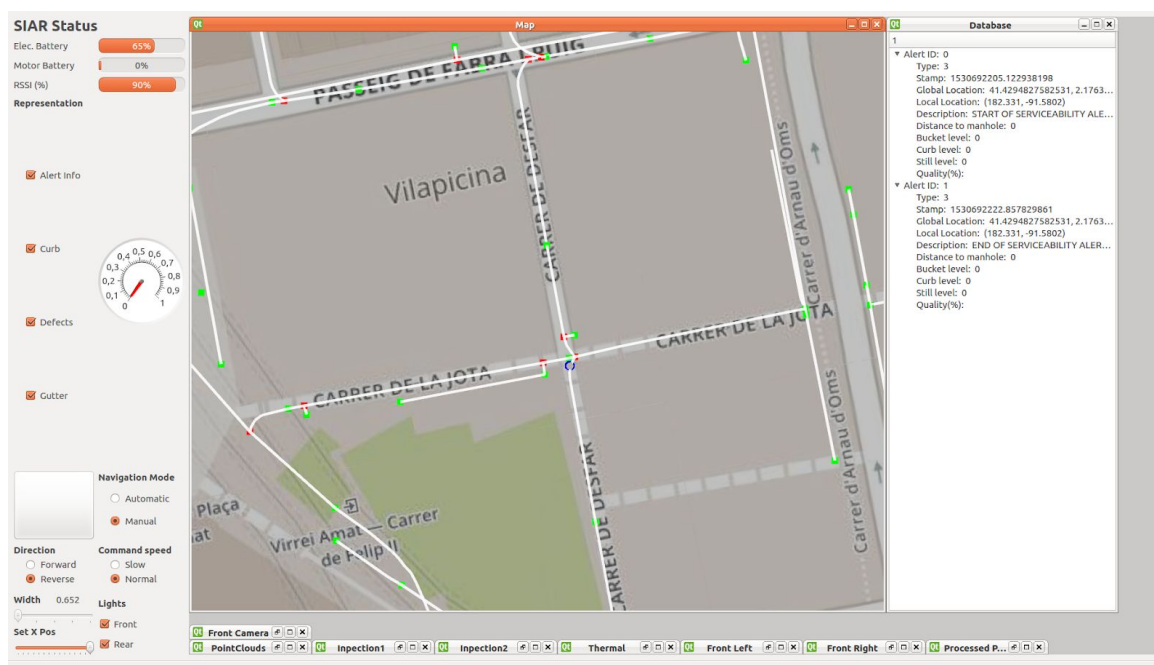


Figure 4.4: Mission execution view of the control station.

The current software has been successfully tested during the experiments carried out in three different areas: *Virrei Amat Square*, *Av. Pearson* and *Passeig Garcia Faria*. In particular, during the experiments, different operators from BCASA were capable of guiding the SIAR platform in a straight section of the

sewer from the outside of the sewers in assisted teleoperation mode for more than 200 m. Figure 4.5 displays a picture taken during one of these experiments.



Figure 4.5. An operator from BCASA safely guides the robot in semi-autonomous mode from the surface.

4.2. Details on the proposed communication middleware

The communication system of the SIAR platform consists on a link, which operates following the WiFi 802.3b,g standards over the 2.4 GHz. To make a better use of the bandwidth of this link, we decided to develop a custom middleware over UDP for real-time communications. A comparison with the standard ROS middleware can be found at D28.6, Section 4.2. In this section, we will highlight the new functionalities added during Phase III.

Figure 4.6 shows the block diagram of the proposed middleware solution. In this solution, the ROS middleware is employed to connect the internal nodes on both the onboard NUC computer and the base station.

Then, the *udp_bridge* ROS package has been designed and implemented in order to transmit the topics of interest between the robot and the base station through the communication link. The communication is done on a client/server style: the server node runs on robot computer while the client node runs on the base station computer. These nodes can be configured to transmit the desired topics of interest at different rates on request by the operator.

Finally, it is important to mention the modules that are being used to transmit real-time RGBD images to the operator. As mentioned in Section 2.1.2, seven RGBD cameras are being used together to gather the information of the environment. However, in some cases, the proposed communication link can not manage the transmission of information from all the cameras at the same time. Therefore, a camera switch module has been designed which allows the operator to enable/disable RGB and depth video links and to configure their quality in real time in a straightforward manner.

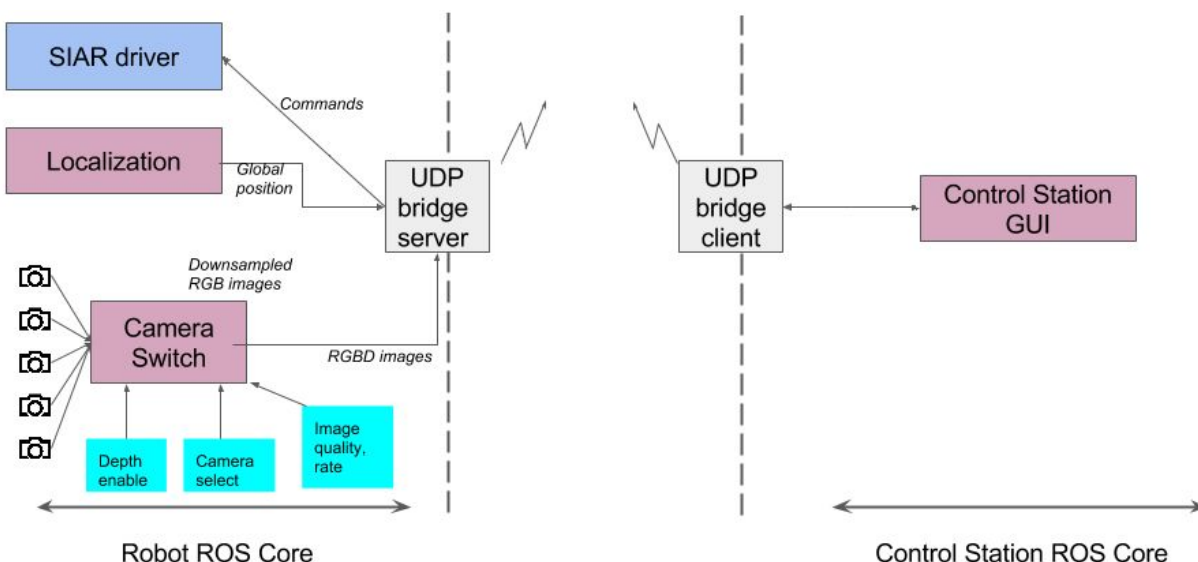


Figure 4.6. Proposed architecture of the middleware of the SIAR solution.

4.3. Localization module

The localization module estimates, in geographical coordinates, the robot's position with respect to the previously existing map of the sewer network. This permits to relate all the inspection results to global coordinates and to integrate them into existing GIS.

The localization module only requires as initial parameters the GIS data in the format provided by BCASA, and the indication of the initial position of operation of the robot. It will then automatically track the position of the robot.

The localization mode is based on the integration of visual odometry, visual detection of manholes and Monte-Carlo localization and the current system is making use of the final version deployed in Phase II. This system has been published as a part of the International Conference on Intelligent Robots and Systems, 2017 [Alejo et al., 2017] and has been described in D28.5 (Section 4.3) and D28.6 (Section 5.2). A brief description of the main functional blocks of the proposed approach is introduced below (see Figure 4.7) for the sake of clarity.

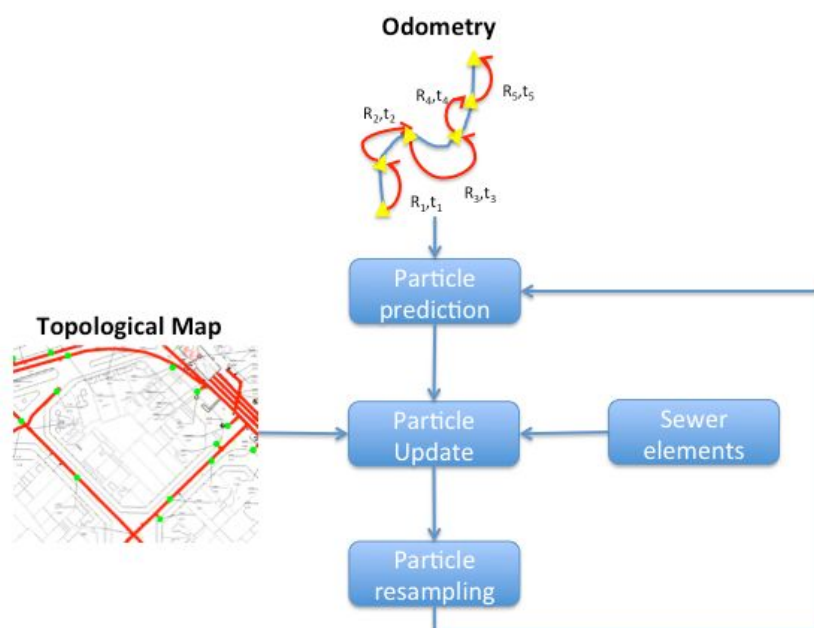


Figure 4.7. Topological map based localization approach. The detected sewer elements, like manholes, and a map of the sewer network including such elements are used to provide a global localization tracking within this map.

- Odometry.** While the robot provides wheel odometry, due to the typical slippage mentioned above, the values are not reliable. Thus, the main odometry source is a visual-odometry method developed for stereo-vision and adapted to RGBD. This block takes as input the RGB and depth flows of the frontal camera of the robot, matches robust features from consecutive or closely spaced frames and then obtains the relative pose between cameras that minimizes the re-projection error. The reader is referred to the D28.1 document, Section 5.1, in which the details on the RGB-D based Odometry can be found.
- Manhole detector.** This module checks whether the robot is under a manhole or not based on the depth images gathered by a camera pointed upwards (please refer to Section 2.2.3 for details on the disposition of the sensors). A machine learning approach is employed to perform this classification robustly and computationally effective. For details on this system, please refer to D28.6, Section 6.3.
- Localization module.** This module integrates the odometry measurements through time, performing proper corrections according to an *a priori* topological map from a GIS and whenever a sewer element is detected (in this case by the manhole detector, but actually any element in the map could be used if it can be detected). A Monte-Carlo localization system has been used (see D28.6, Section 5.2.1).
- GIS information extractor.** This module gets the useful information of the GIS data given by BCASA and generates a sewer graph that can be loaded by the localization software. Please refer to D28.6, Section 5.2.2.

4.4. Navigation module

The navigation module is in charge of the motion and decision making of SIAR. The module offers the capability to be controlled by the operator at all times through the communication system, so that skilled operators can command the robot if required. However, the SIAR system philosophy is to let the robot solve the navigation so that the operator can focus in the inspection task, even if some sections of the sewer are difficult to navigate.

Two main navigation modes are offered by the navigation module:

- **Teleoperation.** In this mode, the robot can be purely teleoperated from the base station, where the operator controls all the degrees of freedom of the robot. This mode has been already implemented and tested for the initial version of the robot.
- **Semi-autonomous navigation.** In this mode the robot automatically navigates through the center of the sewer, avoiding falling into holes, etc, while the operator provides high level commands: move forward, move back or stop. To this end, this module gets a partial 3D reconstruction of the environment from the mapping module, in order to detect the safe navigation areas where the robot can go through.

Therefore, in the semi-autonomous mode, the robot is able to locally follow commands when traversing the linear sewers without falling into the gutter, negotiating obstacles if possible, and avoiding getting stuck. The local navigation module (Section 4.4.1) takes care of this part. The robot should be able to negotiate forks and bifurcations, which typically requires the capability of performing complex maneuvers. The method used in these cases is explained in Section 4.4.2.

4.4.1. Local Navigation

The depth images obtained by the front and rear cameras (six in total) are combined and processed in order to obtain a 2.5D map of the environment close to the robot, storing the height values of the surroundings. This is done in real time at a frequency of 10 Hz. Then, this elevation map is further processed to define the areas that can be traversed by the SIAR platform (see Figure 4.8 right). To date we have empirically set a threshold of 7.5 cm as the maximum step that can be safely negotiated by the platform and a threshold of -7.5 cm that identifies the negative obstacles, which are marked in red. Moreover, the parts of the maps are given a safety penalty which decays exponentially with the distance of the closest untraversable area. This way, the robot can be aware of the parts of the floor that are more dangerous and will avoid them as much as possible.

This traversability map is then used to compute safe trajectories when the robot operates in semi-autonomous mode. The local navigation module gets the desired high level command (move forward, move backwards and stop) given by the operator and then evaluates similar commands. To this end, the module predicts the trajectory of the robot during the next T seconds and checks whether the wheels of the robot will pass through an untraversable zone or not. In that case, the checked command is discarded. Figure 4.8 (right) depicts the footprint of the wheels of the robot for the tested predicted trajectories by the module.

The rest of safe trajectories that are being tested are then ranked according to the following formula.

$$J = \omega d + \int_0^t P(t) dt$$

where ω is the command weight that multiplies the difference d between the operator's command and the command being tested. $P(t)$ is the safety penalty obtained when the wheel footprint is applied to the altitude costmap at time t . In all the tested cases, ω was set to 0 and thus the navigation module always selected the safest trajectory.

Finally, the trajectory with lowest rank is kept and its related command is sent to the controller of the robot. The best trajectory is marked in blue in Figure 4.8. This operation is also repeated at a frequency of 10 Hz. The local navigation module has been successfully tested in the real sewers in several places of Barcelona during experiments in 2017 and 2018.

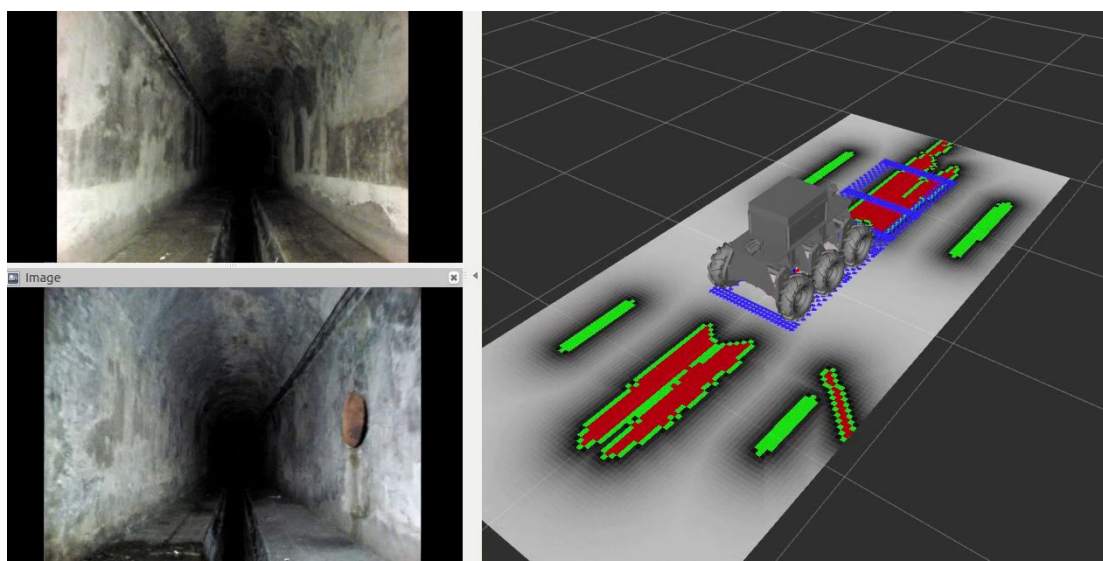


Figure 4.8. Left: Picture obtained from the operator awareness camera. Right: Associated altitude map (positive obstacles are marked in green, negative obstacles in red, traversable areas in gray and generated trajectory in blue).

4.4.2. Navigation module in the presence of forks

Even though the local navigation module has proven its reliability in the previous experiments, by safely traversing hundred of meters in areas of the sewers without forks, it is not designed to be able to traverse a bifurcation. The reason is quite simple: when the robot has to traverse a bifurcation some of the wheels should be allowed to enter the gutter during the maneuver. Unfortunately, the local navigation module is based in its reliability in not letting any wheel to enter the gutter. Nonetheless, in Phase II, the automatic control method described in Section 4.4.1 was adapted to cross fork and was successfully tested in real experimentation (please refer to D28.6 Section 5.3.2).

To improve the robustness of the module, we have developed a new software module that implements a planning algorithm to automatically generate a sequence of commands that allow the robot to safely cross a fork. Moreover, we have tested the proposed planning in different scenarios obtaining very satisfactory

results: the proposed algorithm was able to obtain feasible solutions within few tenths of a second in average.

Depending on the environment where this proposition is studied, there are different approaches that can be adopted. In our instance, we analyze the case of a terrestrial platform moving in a sewer with narrow corridors and forks where the path is divided into different passages. In addition, it also presents a conduct or gutter on the floor that is used to drain the waters, as seen in Figure 4.9.

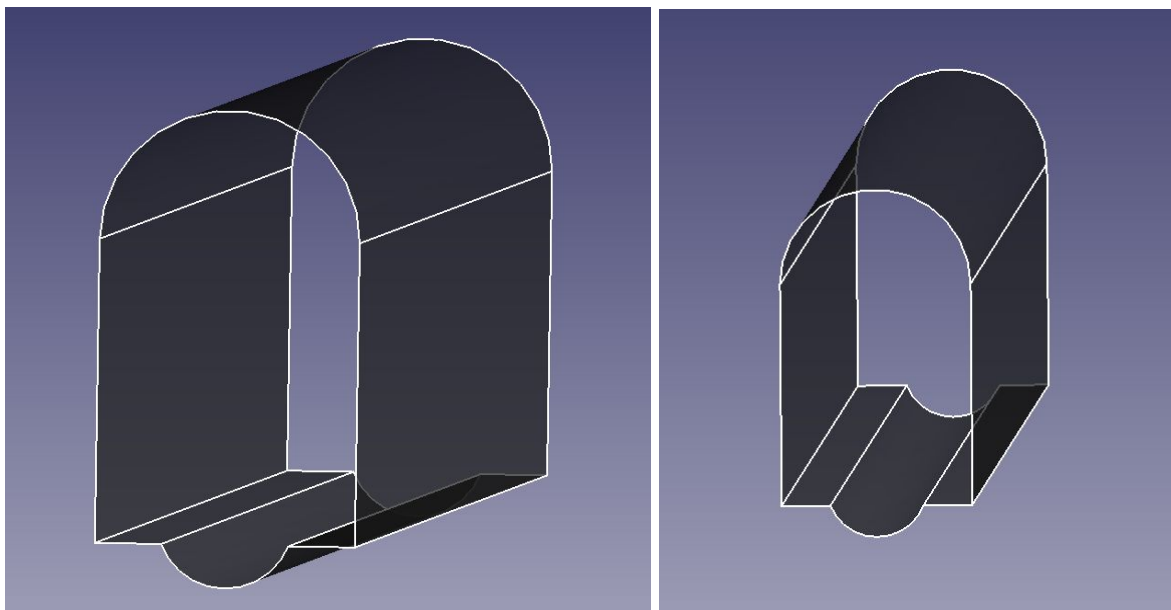


Figure 4.9. Two examples of sewer sections considered in the planning algorithm.

Based on these conditions, we may distinguish two types of obstacles to avoid. Positive obstacles which are those that stand over the floor, such as the walls or variety of rubble it may encounter. And negative obstacles that stay below, such as the central gutter. We will consider a collision when the platform navigates over any of these obstacles, but only when they exceed a given range that make navigation impossible or dangerous for the platform. To identify the obstacles, we make use of the same obstacle map that is described in the Section 4.4.1.

We use two types of algorithms to provide the platform with autonomy. On the one hand, we can obtain an optimal global trajectory to move between two distant points inside the sewer. In this case, we use the graph of the sewers provided by local authorities and obtain the best path by using the well-known A* algorithm.

However, we have to generate safe commands for the platform to make it able to follow the global path obtained by the A* algorithm. This procedure is usually named local planning. For local planning, we have developed a kinodynamic RRT algorithm [LaValle et al, 2001] that shows much better time performance than the A*, which was presented in D28.5, Section 4.1.2. Even though, RRT does not achieve the most efficient path, as A* does, the computational time savings justify its use. Furthermore, as we are planning into a confined and narrow environment, obtaining an optimal trajectory is as critical as obtaining a safe trajectory.

The planning problem is defined with a starting and a goal point (q_{start} , q_{goal}), which are obtained from the global trajectory. Then, this algorithm uses a random sample of the search space and makes a tree of configurations to grow from this start point, until getting to the goal at any moment.

The process is the following: on each iteration, we choose an aleatory pose of the available space, this is q_{rand} , as depicted in Figure 4.10 (left). Once this random pose is selected, we must find the closest node of the already existing tree to this q_{rand} , what we call q_{near} . From q_{near} as a parent node, we will grow the tree, making it to tend to q_{rand} . This last step is done by establishing a fixed growing step epsilon, that will lead us to get a new pose called q_{new} . We use a Kinodynamic RRT to make sure that the generated trajectory is feasible. In this case, we integrate in the vehicle model a fixed amount of time (T) to calculate the commands from a parent node to the new model. To do so, we test an amount of K random commands and we keep as q_{new} the closest point to q_{rand} .

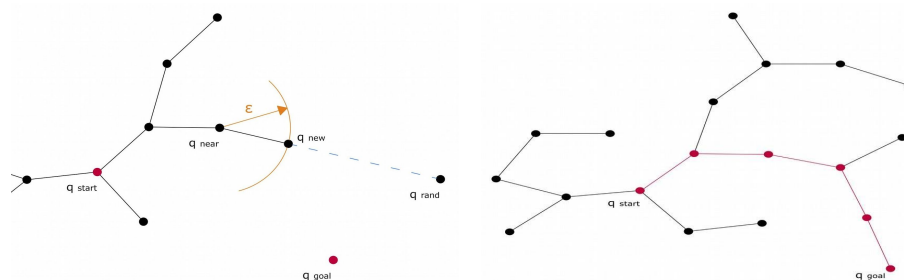


Figure 4.10. Left: RRT extend step. Right: final tree and solution (in red) of an execution of the RRT algorithm.

As this algorithm tends to cover the full configuration space, at one point q_{new} will be closer to the goal than a threshold (goal gap). This is shown in Figure 4.10 (right).

We have found that we obtain better results when applying a well-known variety of the RRT: bidirectional RRT (biRRT). This is an alternative RRT version where two different trees are created simultaneously from both sides, the starting and goal points. At some time, these two trees can be connected to each other, giving place to a single path that is the solution. By using this version of RRT we avoid the complexity that implies the presence of a fork when selecting the right path to the goal. When both trees get to the bifurcation, they can find each other easily, and connect to each other much faster, as shown in Figure 4.11.

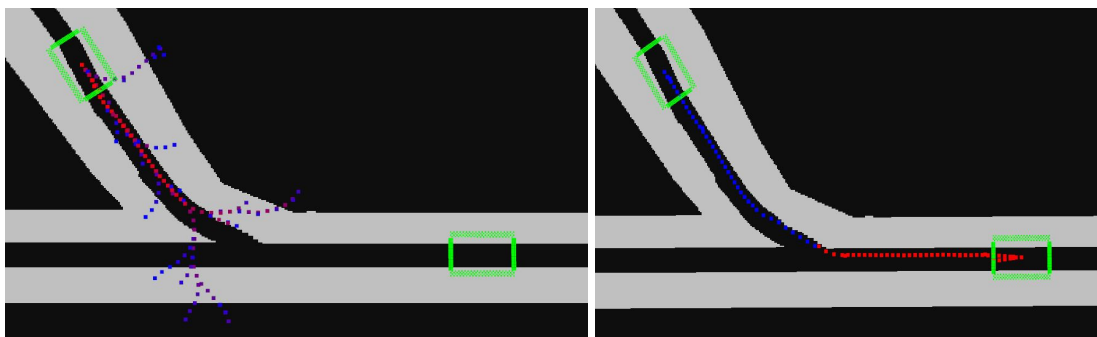


Figure 4.11: Comparison of the RRT (left) and bi-RRT (right) planning algorithm when applied in a forked scenario.

In addition to the random sampling of the space when creating the RRT tree, our algorithm can select the goal configuration instead of the random configuration to encourage the growth of the tree towards the goal with a determinate rate (goal bias). In biRRT case, the random configuration is defined as the middle point between the closest points of the two trees.

Apart from the goal bias, there are other parameters that should be carefully selected to improve the performance of our implementation. Among these, the most important ones are: the maximum number of iterations (n_{iter}) to achieve the goal, the number of commands when generating q_{new} (K) and the integration time (T).

On the following table, the results of the different planning algorithms tested under the same environment conditions and varying the parameters for RRT and biRRT, are shown. This show differences when changing these parameters with regard to the base case (K = 20, goal bias = 0.1, n_{iter} = 200). We keep the rest of parameters, such as integration time step (T = 0.2) or sampling space fixed for all cases.

100 Tests per case		Average execution time (s)	Av. Success Time (s)	Av. Failure Time (s)	Failure rate (%)
Forked Scenarios	biRRT Base Case	0.58	0.40	1.36	19
	biRRT K=5	0.13	0.11	0.40	10
	biRRT goal bias = 0.33	0.76	0.50	1.43	28
	RRT Base Case	0.64	0.40	0.70	80
Not Forked Scenarios	RRT goal bias = 0.02, K = 5, n_{iter} =2000	1.69	1.11	2.05	62
	biRRT Base Case	0.24	0.21	1.28	2
	biRRT K = 5	0.07	0.07	0.00	0
	RRT: goal bias=0.2, n_{iter} =500, K=5	0.25	0.11	0.48	39

Table I: Obtained results with the proposed planning algorithm in multiple runs.

As we can see from the results, biRRT always gives us the best time performance and best success percentage. This becomes more evident in forked scenarios, where normal RRT performs significantly worse due to its huge failure percentage. This failure rate can be decreased by increasing n_{iter} , but it implies an increase in the execution time.

In conclusion, we have implemented and tested a new planning algorithm that allow us to perform fork negotiation. Moreover, its low execution time allow us to use the algorithm in a local replanning module in those cases.

4.5. Sewer Map Building

The sensorial system of the robot provides 3D scanning data. These scanning data can be accumulated in time to build 3D **local maps** of the sewer, which are the basis for automatic inspection of the sewer. These local maps can be used to analyze the sewer serviceability, as will be described below. The 3D reconstruction can be also compared with pre-computed 3D models of the gallery in order to detect possible defects.

The 3D local map is also very important in order to provide environment awareness to the teleoperator. The operator has a first-person perspective of the environment thanks to the cameras onboard the robot. Sometimes it needs a third-person perspective in order to fully understand the robot with respect to the scene. Figure 4.12 shows a snapshot of the information presented to the operator. It is possible to see that the images are good to detect possible defects or to understand the operation, but the third-person perspective provides a global view of the robot position that contributes to understand how the robot interacts with the scene.

Furthermore, the 3D scans can be also used to build a **global 3D map** reconstruction of the sewer, typically offline once a mission has finished. This can be used to update old maps, mission analysis, etc.

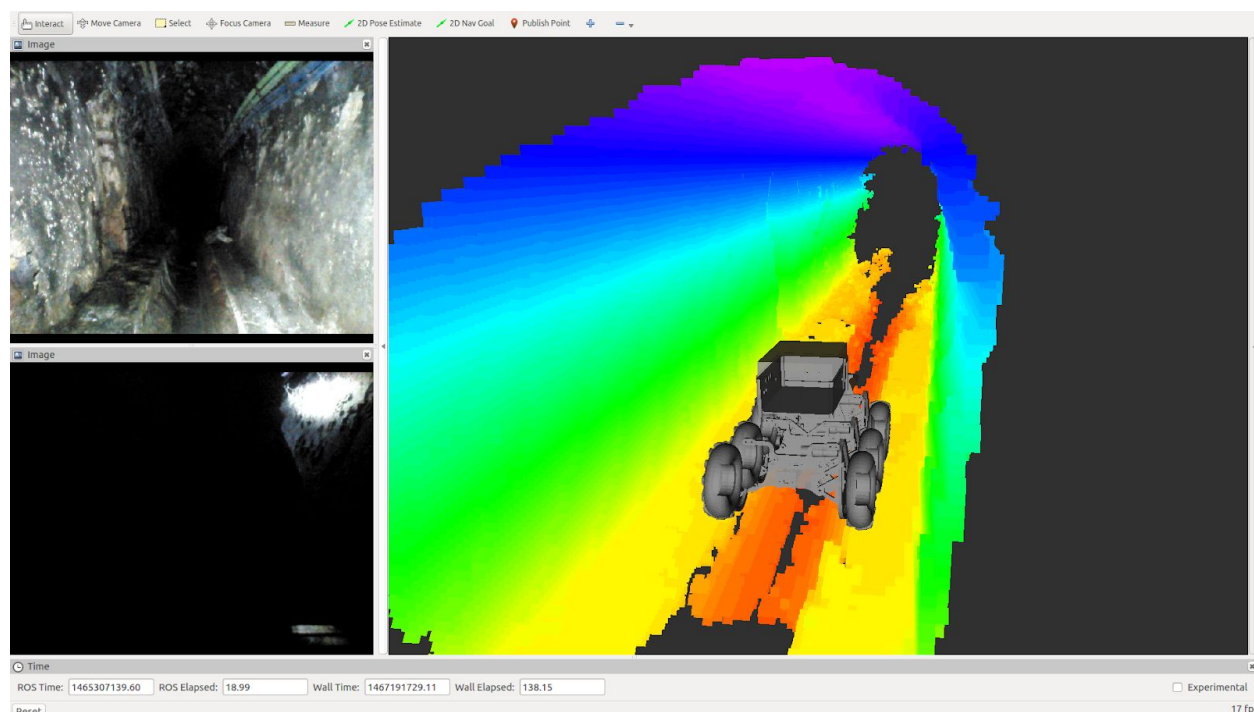


Figure 4.12. Different views of the operator while commandind the SIAR platform.
Left: first-person perspective (front and rear cameras). Right: third-person perspective.

Finally, these maps, combined with the localization outputs (Section 4.3), can be georeferenced into the sewer network.

Next subsections provide some details about the 3D local and global mapping process, together with some experiments carried out in Barcelona to validate the approach.

4.5.1. Local maps

The main purpose of the 3D local mapping module is to provide a locally coherent 3D map of the robot environment. For this purpose two elements are used as inputs: 3D point-clouds provided by RGB-D cameras mounted in the robot and an accurate robot odometry for the integration of the point-cloud information through time.

The local map is represented as a 3D occupancy grid that stores the probability of the cell of having an obstacle. This 3D occupancy grid has a predefined resolution of 25 mm and a fixed size around the robot that can be adjusted depending on the requirements. The occupancy grid is implemented as an octomap [Hornung et al, 2013] in order to make a very efficient use of the computer memory. The local map is created with respect to a particular keyframe.

The estimations from the odometry system detailed in Section 4.3 are used to align the point clouds with the current map so that it can be merged with it. The motion of the robot is checked in order to detect a new keyframe position. If the robot moves more than a given threshold, a new keyframe is added and all the map points within a distance are translated to the reference of the new keyframe. This method allows to reduce the computational requirements for map building and map representation.

This system was already deployed in Phase I and described in detail in D28.1. Figure 4.12 shows one of such local maps.

4.5.2. Global 3D maps

The local maps described above can be used online for serviceability inspection, situation awareness, etc. We have extended this functionality to be able to also provide full global 3D maps of the inspected area offline. The approach has been described in detail in D28.5. We summarize it here.

The SIAR approach for global 3D reconstruction is based on the application of an off-line robot trajectory optimization. It considers the information provided by the sewer elements automatically detected by the robot to correct the errors of the odometry and 3D local maps described above.

Thus, the approach builds a robot pose graph based on the motion provided by the visual odometry system (Section 4.3). Due to the lack of global information, the robot position estimation will drift with time, and will eventually diverge. The objective is to simultaneously estimate the robot's position and also to 3D map the environment, based on RGB-D data gathered by the robot.

A pose-SLAM system based on off-line nonlinear optimization has been implemented. This system tries to determine the robot poses that minimize the errors on the following elements:

- The relative position between consecutive poses given by the visual odometry;
- The position of the manholes automatically detected and matched;
- The alignment of the 3D point-clouds gathered by the robot in each pose of the graph.

The outcome of the previous optimization will be a globally consistent trajectory. After the pose optimization stage, the point-clouds of every single pose can be projected into world reference frame in order to build a globally consistent map of the robot trajectory.

Figure 4.13 shows a 3D reconstruction of a sewer from one of the experiments carried out. It can be seen how the general structure of the sewer section can be visualized. This 3D reconstruction is metric, so that the operator can take measurements of the different areas of interest in the environment.

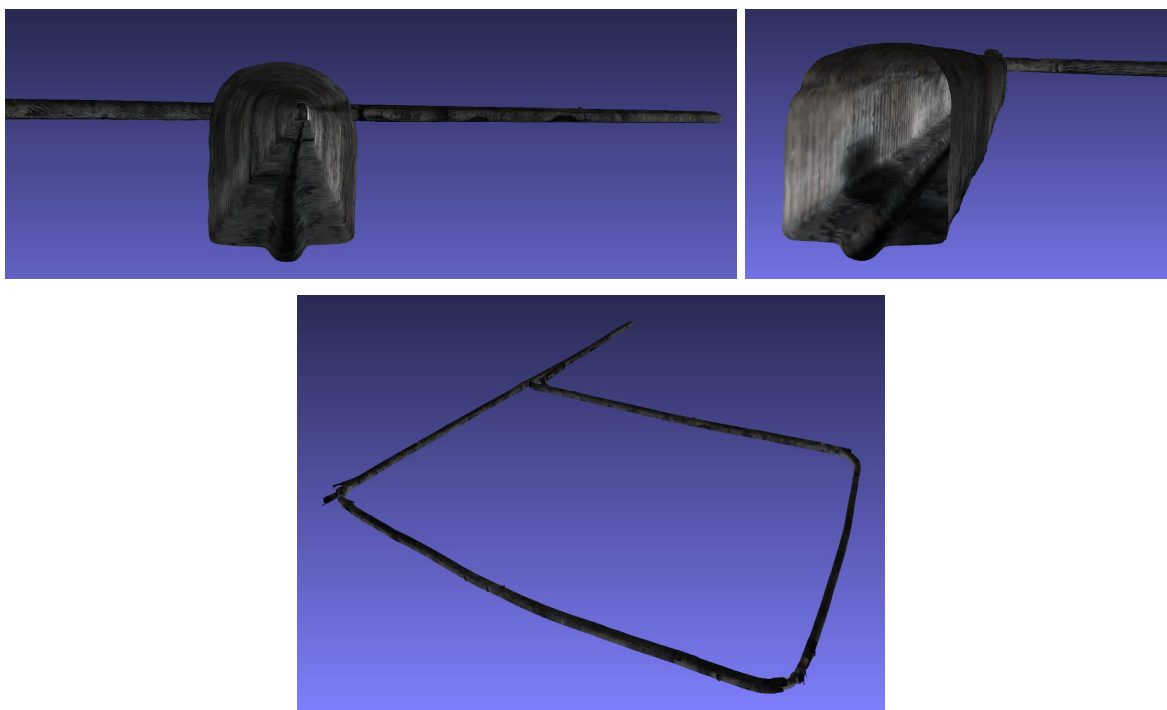


Figure 4.13: Two close-ups and a global view of one of the global maps obtained.

4.6. Map Analysis

The previous modules are general, and can be employed for different applications of the robot. The Map Analysis module implements the inspection functionalities required for the particular application of sewer inspection.

The module has been designed according to the requirements established in the Challenge Brief [ECHORD++, 2014], and refined in meetings with BCASA.

The module offers two main functionalities: Serviceability Inspection (described in detail in D28.10), and Structural Defects inspection (described in D28.12). Here we summarize the module (and new results are presented in Section 5), and refer the reader to those documents for further details.

The module uses the 3D input data given by the set of cameras of the robot and integrated by the mapping modules, and generates, in real-time, serviceability and/or structural defects alarms that can be displayed in the Control Station for the operator. Furthermore, those alarms are time-stamped and geo-located, and a report is generated when the mission finishes for mission de-briefing and post-processing.

Figure 4.14 summarizes the processing pipeline. The proposed system follows the following strategy, indicated in the Challenge Brief, for both, serviceability and structural defects inspection:

“On the basis of the scanning or the video made, the robot has to compare the obtained data with the available information of the sewers (mainly type and section) and identify where the sewer serviceability

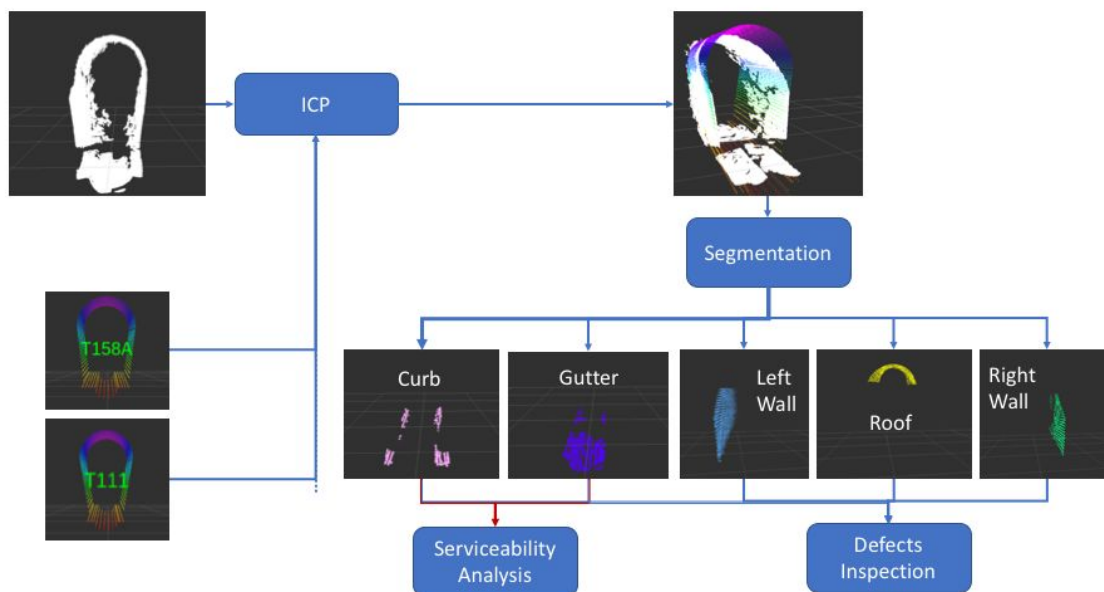


Figure 4.14: Main processing pipeline. ICP is used to estimate (and align) the current section type by comparing the 3D data (in white) with virtual models from the database. Then, the parts (curb, gutter, walls, roof) are segmented from the input point cloud. These parts are analyzed to estimate potential serviceability alarms and defects.

has been reduced. The operator should receive a “pop-up” alarm that indicates the location of the obstruction and helps to decide if the robot has to make an extra specific snapshot or video.”

The system measures the deviation of the online 3D data from the ideal situation to raise alarms. Noise reduction and temporal consistency filters are then applied to discard false alarms.

The main steps are thus, the following:

- The robot automatically recognizes the traversed section type by using the 3D data provided by the robot sensors. It does this by comparing these data with the known section types in a database, which can be used to obtain a virtual 3D model of the sewer section. Information from the robot localization system can be also used to estimate the section currently traversed.
- Once the section type is determined, the different parts of the sewer (gutter/bucket, sill, curbs, roof and walls) are identified in the 3D data.
- For serviceability, system analyzes the gutter/bucket, the sill and the curbs to determine potential serviceability reductions, according to the specifications. It estimates the current free space for the flowing of residues (see Fig. 4.15).
- The roof and walls are analyzed for structural defects, by comparing the 3D model of the sewer section and the actual 3D data from the sensors (see Fig. 4.16).

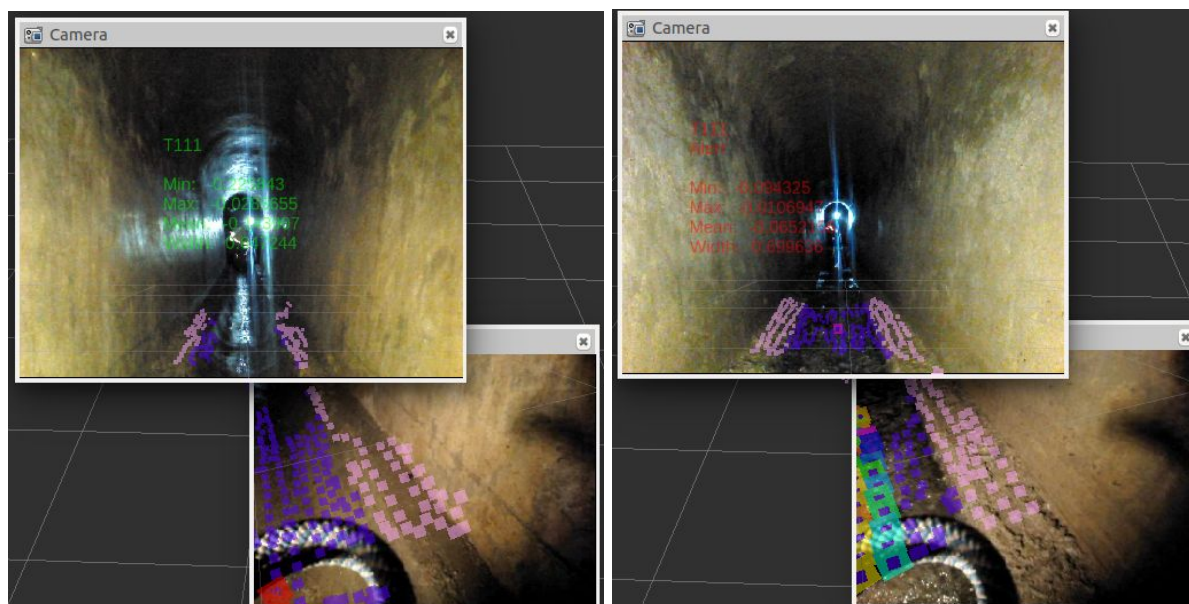


Figure 4.15: The frontal and down right cameras are shown (there is a down left camera that is not shown for clarity). The section type and the minimum, maximum and average values for the height of the gutter are displayed in the frontal image for the operator (a minimum of -0.25 means that the gutter reaches 25 cm. below the robot). By analyzing these values it is possible to estimate the serviceability level. In purple and pink, the 3D points corresponding to the gutter and curb respectively. Left: the serviceability of the gallery is correct. Right: serviceability alarm. The values are displayed in red. Many of the 3D points in the down-right camera (marked as colored points) are detected as departing from the ideal section (the gutter is blocked by debris).

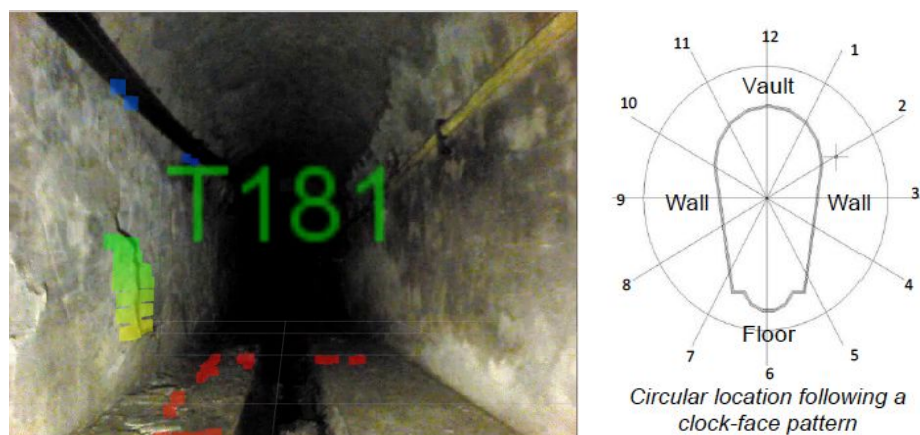


Figure 4.16: Defects detected using data from Phase II experiments. Left: the base station highlight a part of the image that deviate from the section model automatically estimated (T181 in this case). Right: as the 3D position of the points are known, it is possible to indicate the location of the defects following a clock-face pattern.

The alarms can be used by the operator to use the robot arm and camera to closely inspect the scenario, as shown in Figure 4.17.



Figure 4.17: Two images obtained from the arm camera. Left: Detail of an obstacle. Right: detail of a wall.

5. Report on the final demo of December, 13th

The whole system as described in Sections 3 and 4 has been tested extensively during Phase III. Several reports regarding to the performed tests can be found in D28.9, D28.10 and D28.11.

In this section, we provide some results that prove the correct behavior of each component of the system with the data acquired in the experiments of the Final Demo of Phase III, which was carried out in December 13th, 2018.

5.1 Inspection plan for *Passeig de Garcia Fària*

Figure 5.1 represents the inspection plan generated for *Passeig de Garcia Fària*. The following points detail the performed tasks for the inspection of this area.

1. **Robot deployment.** The robot will be deployed in a manhole nearby the center of the area to be inspected. By doing so, we maximize the areas with direct communication with the robot and thus, the performance of the communication system. However, due to constraints in the location of the demo, the base station was deployed near one of the endpoints. The communication device was installed in one of the steps of the manhole. The deployment time is done in 15 minutes on average.
2. **Repeater deployment.** Once the robot and the communication equipment of the base station have been deployed, the operators should then deploy the repeaters in the locations signalized in Figure 5.1 corresponding to the manholes MH109 and MH179 in Figure 5.2. The operator at the base station should then confirm the connectivity with each one of the deployed repeaters. The time for deploying the repeaters is about 2 minutes for each repeater, plus the walking time. In total, the operation can be done in less than 15 minutes. Note that this has been improved with the use of the new self-deployable devices described in Section 3.1.
3. **Connectivity checks.** Then, the connectivity with the robot and the deployed repeaters should be checked in the base station. Also integrity checks of the software modules running onboard the robot are performed. This can be done in 5 minutes on average, but can be performed in parallel with task 2.
4. **Sewer inspection.** The setup is concluded and the operator is able to perform the inspection over the *Passeig de Garcia Fària*. The expended time to inspect a section depends on the length of the track, the type of section to be visited and the number of defects found during the inspection. An estimation of this time can be obtained by using Eq. (1). In this case, in the section type of the track T130, the robot was commanded to proceed with a width of 72 cm, which prevents the platform from falling inside the gutter. It has a length of about 400m. This gives an expected inspection time of 1 hour and 30 minutes. However, due to the weather conditions, the mission was aborted at the half of the track: only 200 meters were inspected in 45 minutes.
5. **Robot retrieval and cleaning.** The robot should be retrieved from the same manhole where it was deployed. Once the robot is collected, it should be cleaned with sprayed water. In the meantime, the other operators can collect the deployed repeaters. The estimated time to perform this operation is about 20 minutes.

The following formula can be applied to estimate the inspection time of a linear segment:

$$t = \frac{2L}{0,5(v_i + v_b)} \approx 10L \quad (1)$$

where L is the length of the track, v_i is the inspection velocity (approximately 0,15 m/s) and v_b is the velocity when returning from the track (approximately 0,4 m/s). The expression can be approximated to just ten times the length to be inspected (L), this assumes that the average speed to perform the track in both directions is 0,2 m/s.



Figure 5.1: Inspection plan for *Passeig de Garcia Fària* area. The black boxes indicate the position of the repeaters and the robot points to the deployment manhole.

5.2 Localization results

Figure 5.2 represents the estimated trajectory executed by the SIAR robot in the experiments of December, 13th. In this particular case, the implemented localization algorithm had an excellent performance: the red line that represents the estimated trajectory follows strictly the topology of the network and the robot passes below the manhole where Repeater 1 had been deployed: MH 109.



Figure 5.2. Localization results of the robot's position in the experiments of December, 13th.

To validate the precision of the results, we compared the estimated position obtained by the system with the position of the manholes given by the provided GIS information. The distances of these positions are detailed in Figure 5.3 in multiple runs of the proposed algorithm. As expected, these distances do not exceed 1.5 m for most cases in the experiments when considering the manholes (MH) numbered 102, 104 and 110. However, there is a systematic error of more than 7 m that is always found when considering MH 109. This error suggests that the GIS information is wrong when considering this manhole. Thus, the position of the manhole should be updated and our system estimates that the new

position should be of 41.4062216401175°N, 2.21707423186689°E with an accuracy of 2m, approximately (see Figure 5.4).

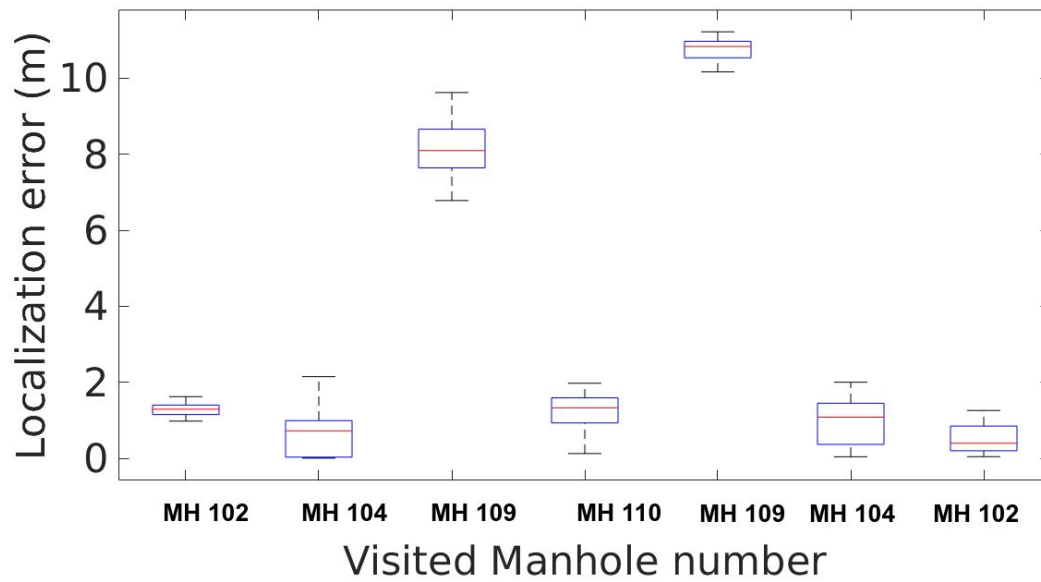


Figure 5.3: Localization error of the localization system when the platform passed below a manhole.

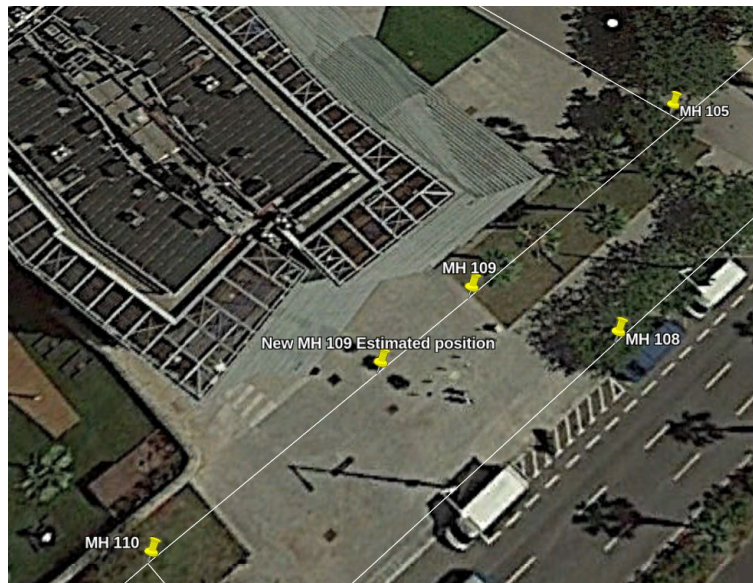


Figure 5.4: GIS position of MH109 and new estimated position of MH109.

5.3 Navigation

The complete inspection procedure was carried out in semi-autonomous mode with the exception of the need to deal with a big obstacle as described in Section 5.5.2. In particular, as no forks were crossed, we tested the algorithm described in Section 4.4.1. By using the proposed system, the robot traveled a total of 400 m in less than 40 minutes, which gives an average inspection rate of 0.18 m/s. Please note that this inspection time includes all the time since the first movement and as such it includes the time where the platform was stopped to take detailed pictures of potential defects as well as the time where the robot was stopped to provide some explanations on the platform and software systems to the reviewers. When the robot was moving from one point to the other, it was moving at a speed of 0.5 m/s most of the time.

5.4. Communication experiments

In this section, we analyze the performance of our new self-deployed repeaters in the sewers of *Passeig de Garcia Fària*, where the final tests of Phase III were carried out. This deployment mode of the repeaters was used to successfully tele-operate the robot in semi-autonomous mode from the surface, during the whole track. The image flow provided to the operator during the full experiment was stable.

Furthermore, the proposed configuration allowed an operator from BCASA to tele-operate the robot from the base station in semi-autonomous mode in one of the preparation experiments for the final demo. He was able to bring the platform home from the furthest point of the track. The operator did not receive any prior training for doing this, only a small briefing was given before commanding the robot.

5.4.1 Experiments at *Passeig de Garcia Fària*, December 2018

Three wireless communication systems (one at the base and two repeaters) were deployed in the sewers in three manholes by fixing them to the stairs.

Figure 5.1 represents the position of the repeaters. The robot was moved through the inspection course and it collected the received RSSI of each device. The collected information was used to produce the coverage map, in which the highest RSSI of the deployed devices is represented as a function of the location of the SIAR platform.

Regarding to autonomy, the repeaters were continuously used during several experiments which in average lasted 4 hours. They were discharged less than half of their capacity during these experiments. Therefore, they have an adequate power autonomy for a complete inspection procedure.

Figure 5.5 shows the RSSI power of the signal of each repeater as perceived by the communication system onboard the robot as a function of the distance traveled by the robot. As expected, the power of the signal of the Base Station (red line) decreases as the robot performs the inspection. Moreover, the power of the signal of Repeater 1 grows until the robot reaches its deployed manhole. Finally, the signal of Repeater 2 grows with the distance, but is received always with less power than the other, as the robot could not reach the manhole where the repeater was deployed.

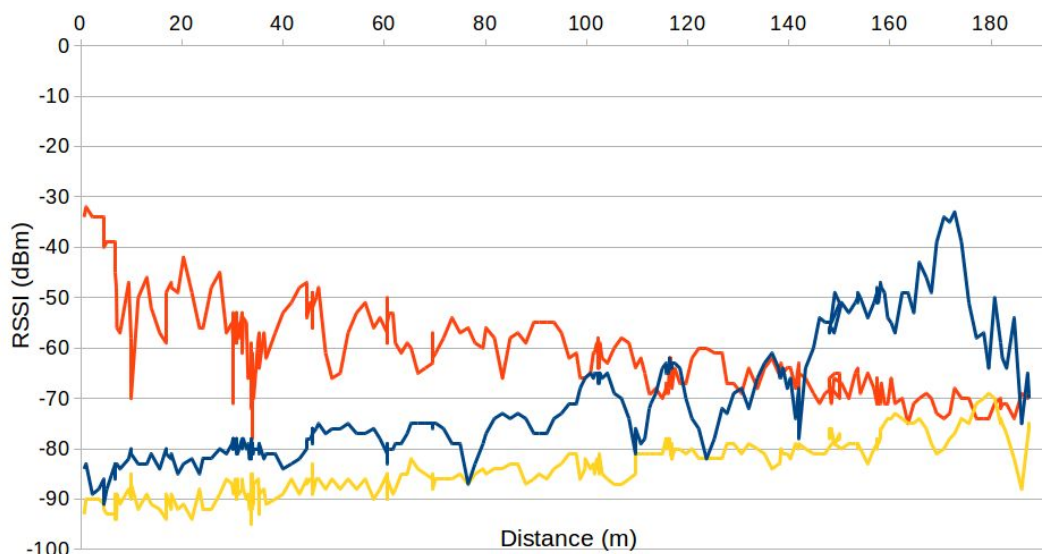


Figure 5.5: RSSI perceived from the communication system onboard the robot of the deployed communication system. The signal from the Base Station, Repeater 1 and 2 are plotted in red, blue and yellow lines, respectively.

5.5. Inspection report

In this section, the defects that have been found by using the proposed automatic defect module and revised by operators are detailed. In particular, we have found three major defects that are marked in Figure 5.6 and are listed below:

1. Serviceability defect 1 (alerts 0 and 1). The module detected an anomaly in the serviceability in the first part of the inspected area. In this case, the water covered the complete surface of the floor and thus the anomaly was reported. The details are described in Section 5.5.1.
2. Structural defect 1 (alert 2). The module detected an obstacle in one of the sides of the gallery. The details are described in Section 5.5.2.
3. Structural element 1 (alert 3). The module detected an inlet in one of the sides of the gallery. The details are described in Section 5.5.3.
4. Structural defect 2 (alert 4). The module detected an obstacle in one of the sides of the gallery. The details are described in Section 5.5.4.

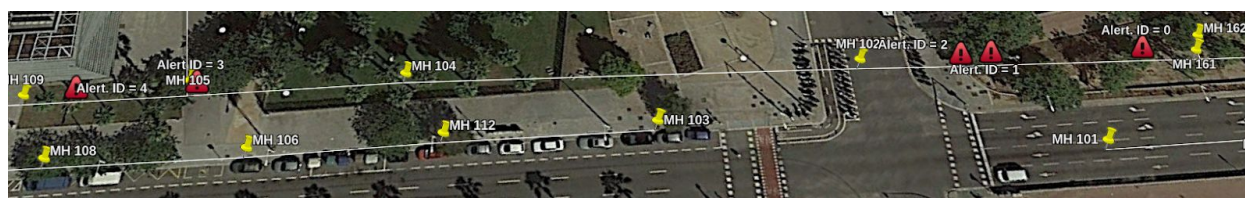


Figure 5.6: Inspection report of the Final Demo of the 13th of December.

5.5.1 Serviceability Defect 1 (Alerts 0 and 1)

Figure 5.7 represents a snapshot of the defects related to Serviceability Defect 1. In this case, there is a level of water over the floor and therefore, the data obtained by the platform indicates an anomaly in the measured gutter levels. A video of the whole defect can be found at the following link:

<https://drive.google.com/file/d/164Z-tyw1-ixd6PzRCXXr-sppoU3az5Ql/view?usp=sharing>

Begins:

Approximate Location: 41.407280967878° N, 2.2185641247187° E

Distance to closest manhole: 7.4 m. Closest manhole: MH 161.

Local Time: Thursday, 13-Dec-18 11:57:47 (local time)

Ends:

Approximate Location: 41.4071441087723° N, 2.21836850938126° E

Distance to closest manhole: 19.3 m. Closest manhole: MH 102.

Local Time: Thursday, 13-Dec-18 12:02:47 (local time)



Figure 5.7: Composed snapshot of three frontal cameras in the Serviceability Defect 1.

5.5.2 Structural Defect 1 (Alert 2)

Figure 5.8 shows a snapshot (left) and a 3D reconstruction (right) of the defects related to the Structural Defect 1, in both visual and 3D representations. In this case, a noticeable obstacle can be found in the rightmost curb. As the 3D reconstruction respects the metric information, the user could perform detailed measures of the obstacle. In this particular, the maximum height of the obstacle has been estimated as 14.4 cm.

Approximate Location: 41.4071163854107° N, 2.21833027827256° E

Distance to closest manhole: 12.2 m. Closest manhole: MH 102.

Local Time: Thursday, 13-Dec-18 11:57:47

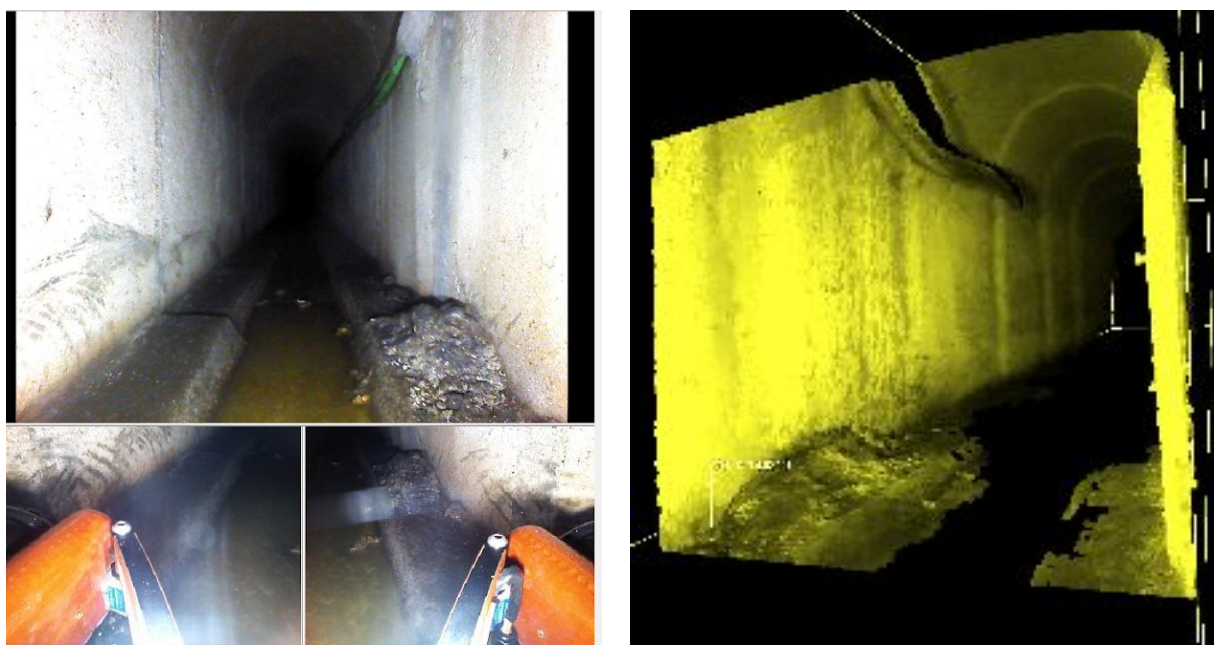


Figure 5.8: Left: Composed snapshot of three frontal cameras in the Structural Defect 1. An obstacle taller than 10cm in the rightmost curb can be found. Right: 3D mesh obtained from the sensors (in the view the obstacle is on the right), where metric measurements can be done (the estimated height over the curb is of 0,144 m.).

A 3D reconstruction of the environment in the surroundings is available on the following link:

https://drive.google.com/open?id=1sNA_I4vdRvNAMC6MhI022gd9tMxjgaQ

5.5.3 Structural Element 1: fork (Alert 3)

Figure 5.9 represents a snapshot of the Structural Element 1, in both visual and 3D representations. This information has been used in order to estimate the height of the inlet, which can be of use to determine if it has been built according to the local laws. In this case the inlet has a height of 19.1 cm.

Approximate Location: 41.4064344080629° N, 2.21736442587566° E

Distance to closest manhole: 1.1 m. Closest manhole: MH 105.

Local Time: Thursday, 13-Dec-18 12:18:17



Figure 5.9. Left: Composed snapshot of three frontal cameras in the Structural Element 1. Some sediments under an inlet over the rightmost curb can be found. Right: 3D mesh obtained from the sensors, where metric measurements can be done (the estimated height of the inlet over the curb is of 0,191 m).

The 3D reconstruction of the environment in the surroundings is available on the following link:

<https://drive.google.com/open?id=1Us67q5dRsFMLEXi6WqMf0UuVfkE1ocLO>

5.5.4 Structural Defect 2: inlet with sediments (Alert 4)

Figure 5.10 represents a snapshot of the defects related to the Structural Defect 2, in both visual and 3D representations. In this case, some sediments nearby another inlet can be observed in the rightmost curb. The height of the inlet is of 7.2 cm.

Approximate Location: 41.4063269886796° N, 2.21721691035362° E

Distance to closest manhole: 8.9 m. Closest manhole: MH 109.

Local Time: Thursday, 13-Dec-18 12:20:01



Figure 5.10. Left: Composed snapshot of three frontal cameras in Structural Defect 2. The defect is found in the rightmost curb and is composed by sediments that come from an inlet. Right: 3D mesh obtained from the sensors. The height of the sediments is estimated in 0,072 m.

A 3D reconstruction of the environment in the surroundings is available on the following link:

<https://drive.google.com/open?id=1ndeSL6YesYmsL0kkGuQNCovEp1imZsLA>

References

- [Alejo et al., 2017] D. Alejo, F. Caballero and L. Merino. “RGBD-based Robot Localization in Sewer Networks”. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, September 2017 (IROS 2017).
- [ECHORD++, 2014] ECHORD++. “Utility infrastructures and condition monitoring for sewer network. Robots for the inspection and the clearance of the sewer network in cities”, Internal Report, 2014
- [Hornung et al, 2013] A. Hornung,. K.M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees" in Autonomous Robots, 2013; DOI: 10.1007/s10514-012-9321-0.
- [LaValle et al., 1998] LaValle, Steven M. (October 1998). "[Rapidly-exploring random trees: A new tool for path planning](#)". *Technical Report*. Computer Science Department, Iowa State University (TR 98-11).
- [LaValle et al, 2001] LaValle, S. M., & Kuffner, J. J.. “Randomized Kinodynamic Planning”. The International Journal of Robotics Research, 20(5), 378–400, 2001.
- [Thrun et al., 2001] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, “Robust Monte Carlo localization for mobile robots,” Artificial intelligence, vol. 128, no. 1-2, pp. 99–141, 2001.