sewer **i**nspection **a**utonomous **r**obot

D28.6 – Robotic Solution Description

SIAR Consortium

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

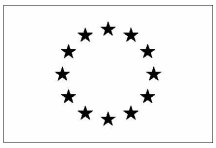

Table of Contents

1. Introduction
2. SIAR Robot Platform V3
    2.1. Robot Features and Components
        2.1.1. Locomotion Platform Main Features
        2.1.2. Sensors
        2.1.3. Actuators
    2.2. Platform Design and Mechanics
        2.2.1. Central Frame
        2.2.2. Width Adjustment Mechanism
        2.2.3. Traction System and Suspension
        2.2.4. Cameras' Disposition and Supports
        2.2.5. Robotic Arm
        2.2.6. Isolation Solutions
    2.3. Robot Architecture
        2.3.1. Electronic Power Architecture
        2.3.2. Communication Architecture
    2.4. Locomotion Platform Electronics
        2.4.1. SIAR Motor control and Linear Motor boards
        2.4.2. Sensor&Management and Battery control boards
    2.5. Improved Kinematics
3. Communications
    3.1. Overview
    3.2. Solution
        3.2.1. Communication device
        3.2.2. Automatic deployment of repeaters
    3.3. Tests and experiments
        3.3.1 Experiments at the Mercat del Born sewers, July 2017
4. Software Architecture
    4.1. SIAR Control Station
    4.2. Details on the proposed communication middleware
        4.2.1. Standard ROS middleware
        4.2.2. Proposed communication middleware
5. Localization and Navigation
    5.1. Enhanced Odometry
    5.2. Localization

5.2.1. Graph-based Localization
5.2.2. GIS information extractor utility
5.2.3. Field experiments in September, 20th, 2017
5.3. Navigation
5.3.1. Local Navigation
5.3.2. Local navigation module in the presence of forks
6. Perception
6.1. Providing 3D scanning data
6.2. Sewer Map Building
6.2.1. Local maps
6.2.2. Global 3D maps
6.3. Sewer Elements Location
6.4. Sewer serviceability inspection
6.5. Structural defects inspection
7. Future Work
7.1. Environmental Sensors
7.1.1. Air Sensors
7.1.2. Water sensors
7.2. Sampling Actuators
7.3. Mechanical Protection
7.4. Automatic defect detection
7.5. Automatic sewer type detector
7.6. Installation of GPU onboard the robot
References

1. Introduction

The SIAR robot platform V3 (version 3) was finished at the end of August 2017. It is an evolution of the SIAR Robot design (version 2) proposed in deliverable D28.1. During this Phase II of the project, the team had the opportunity to test and study different approaches to navigate on the sewers. This and all the information gathered from the feedback of BCASA workers allowed the team to review, change and evolve the proposed design. The new design has been already presented in deliverable D28.5 together with the other developments.

First field tests with V3 robot were done in early September 2017. In these tests, the manoeuvrability of the robot inside the sewer galleries was tested. The tests were performed in Manual, Supervised and Automatic modes. The tests allowed to evaluate the navigation of the robot in straight line, curve and rotating over the gutter and passing or changing direction in the forks. The results were very positive allowing to understand that the new robot has increased locomotion capabilities, over the previous prototype, to navigate inside the sewer galleries.

This document describes the current development state of the SIAR solution and it incorporates updated information from the previous deliverables D28.1 to D28.5.

The document is as organized as follows. Section 2 presents the SIAR robot platform V3, its main features and components, mechanical design, electronics, low-level communication architecture and robot safety. Section 3 details the proposed communication system that will provide the SIAR platform with continuous wireless connection with the base station. Section 4 presents the software architecture. Section 5 presents the autonomous localization and navigation methodology. Section 6 details how the robot will perceive and inspect the environment. Section 7 describes the required logistics and operational issues when using the solution.

2. SIAR Robot Platform V3

The SIAR robot platform V3 is a six-wheeled configuration 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 that are included in the SIAR robot.

2.1.1. Locomotion Platform Main Features

The SIAR robot platform main specifications are:

- Robot Kinematics: 6 wheeled with 6 independent locomotion motors and 1 linear motor for width control
- Weight: 58 Kg
- Battery autonomy: >3.5 hours
- Maximum Velocity: 0.75 m/s
- Acceleration: 1 m/s^2
- Emergency Stop Acceleration: 3,3 m/s2
- Dimensions with maximum width (Height x Max_Width x Length) : 44 x 70 x 84 cm
- Dimensions with minimum width (Height x Min_Width x Length) : 44 x 50 x 98 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 analyzes 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	2
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 + encoder + potentiometer	1

Wireless repeaters deploy system	Quantity
Robotic arm	1
Arm camera	1
Wireless repeaters	3

Sensors and Actuators	Quantity
IMU sensor	1
RGBD Camera	7
Ilumination LED light projectors	2

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

Table 2.1. SIAR devices.

2.1.2. Sensors

The current SIAR robot is equipped with perception and navigation sensors. Environmental sensors will be integrated on the next phase of the project. For navigation the robot uses encoders to control the velocity of the traction motors and an encoder plus 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.

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 45º down and two side-by-side cameras on the rear tilted 45º 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 5-10º up and one on rear tilted 5-10º up.

Perception Sensors

The robot uses the RGBD cameras for sewer monitoring and analysis. The perception sensors are the following:

- 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 45º down and two side-by-side cameras on the rear tilted 45º 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 5-10º up and one on rear tilted 5-10º 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.

2.1.3. Actuators

The robot is equipped with actuators for locomotion, for the deployment/collection of the communication repeaters 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 encoder and potentiometer.
  - Function: change the width between wheels and the centre of mass of the robot.

Communication repeaters deployment/collection system

Wireless communications are quite problematic inside a sewer network. To increase the range of the communications, between the SIAR robot and the remote Control Center, the robot is able to deploy wifi repeaters along its path. For this purpose, the robot is equipped with a robotic arm which will deploy and collect these repeaters. The robotic arm is also equipped with a camera to facilitate the determination of the repeater localization.

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 will be provided by two LED 6W MR16 GU5.3 lamp 6000K projectors (one in the front and another in the rear of the robot). The camera of the robotic arm is also equipped with three high-bright LEDs.

2.2. Platform Design and Mechanics

The SIAR robot is a six-wheeled differential robot with a width adjustment mechanism. The wheel traction system is composed by 6 sets of independent motors, a 90º gearbox and a 260 mm offroad wheel. Each of these sets has a independent suspension arm that connects to the central robot frame. The width adjustment mechanism increases the adaptability of the robot to the sewer configuration. This mechanism is actuated by a linear motor. A central structural frame is used to connect the motorized wheel traction system, the width adjustment mechanism and to carry the payload equipment and electronics. The seven cameras described above are connected to the frame with the use of the camera link supports. These maintain the cameras at the correct height and inclinations. A robotic arm is also connected to the frame.

The arm is used to deploy and collect the wireless repeaters. A fiberglass box was introduced to protect the electronic components from the dirty waters and wastes. Figure 2.1 shows the SIAR robotic platform V3 using the arm to pick up a repeater.

Figure 2.1:SIAR robotic platform V3.

2.2.1. Central Frame

The central frame is made of aluminium and POM (Figure 2.2). The frame accommodates the payload and the width adjustment mechanism. The payload is connected to the upper part of the frame. The linear motor, that actuates the width adjustment mechanism, is connected to the bottom of the frame (Figure 2.3). A solution made of aluminium profiles and aluminium sheets is used to construct the frame. The aluminium profile grants a robust and lightweight solution to the frame. The aluminium sheets are used to confer torsional stiffness. Housings for the bearing that allow the width adjustment are made of POM. The tests performed in the sewer showed a good performance of the frame. No torsion of the frame was detected and all components remain well attached to it.

Figure 2.2: Top and bottom parts of the frame.

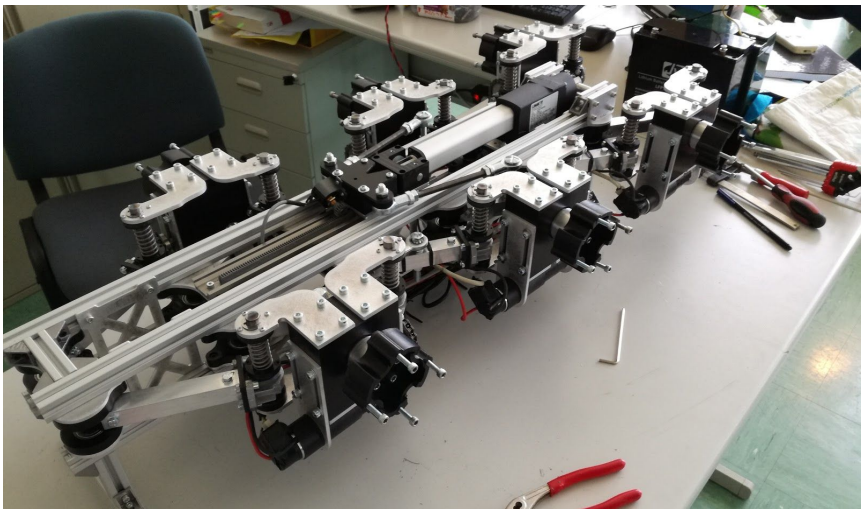

Figure 2.3: Bottom of the frame, linear motor and transmission mechanism.

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 linked to the bottom of the frame and simultaneously to a linear guide. The variation of the stroke of the motor makes the carriage move along the guide. Metallic rods transmit the movement from the motor/carriage to the wheels. Rod ends are used to allow the rotation of the rods at their ends. Each of the six wheels of the robot is connected to the frame by means of two arms. At each end of the arms there are bearings to allow the rotation of the arms and in turn the width variation of the robot. The connection of the arms to the frame and to the wheels generate a parallelogram. The parallelogram assures that the wheels are always kept parallel to the robot. Two main rods transmit the movement from the carriage to the central wheels of the robot (one on each side). Then, on each side of the robot, the wheels are connected to each other by rods, imposing synchronous movement of the six wheels of the robot. Figures 2.4 and 2.5 present two width configurations. With this mechanism the width of the robot can vary from 500 mm (minimum) to 700 mm (maximum). The width variation implies a length variation of the platform. For the minimum width of 500 mm the robot has its maximum length of 980 mm. For the maximum width of 700 mm the robot has its minimum length of 845 mm. The height of the robot is independent of the the width of the robot.

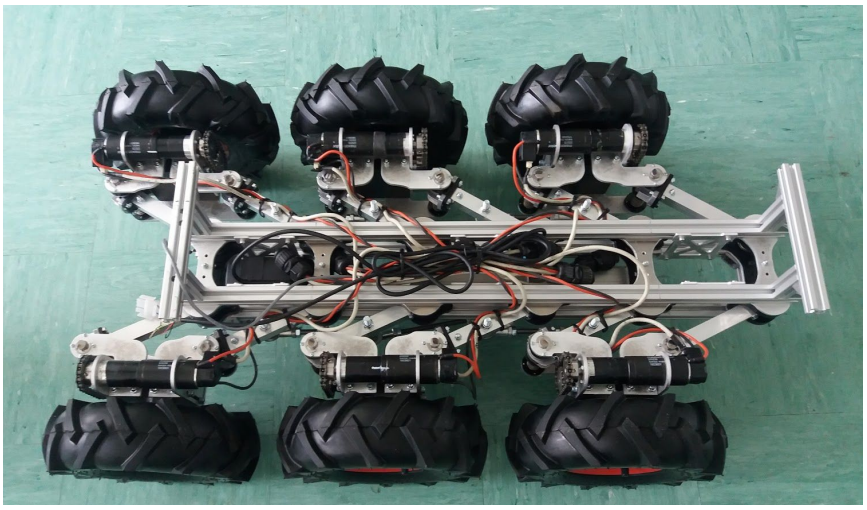

Figure 2.4: Minimum width 500 mm.

:

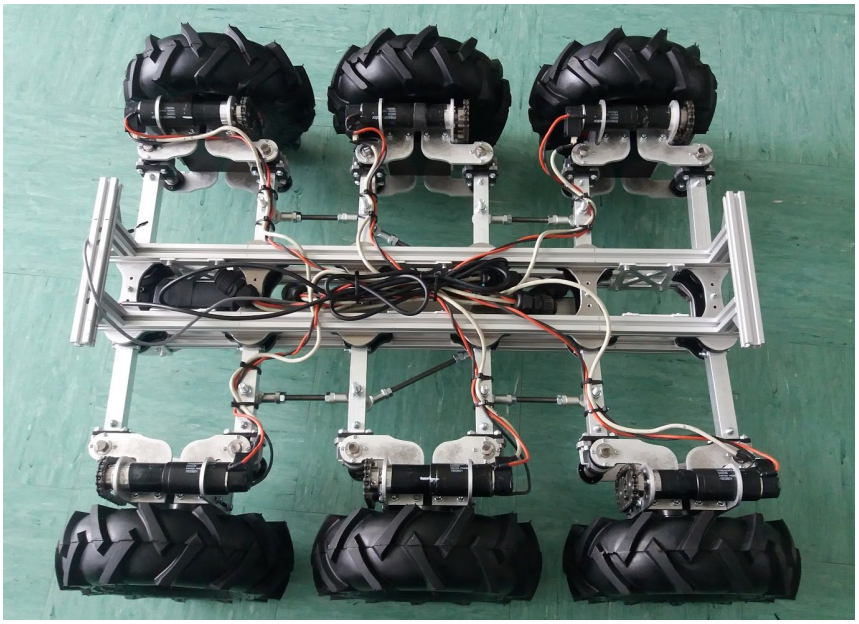

Figure 2.5: Maximum width 700 mm.

The mechanism described above showed 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 center wheel can reach 30 cm (-15 cm to +15 cm).

2.2.3. Traction System and Suspension

As already described, the SIAR robot uses a configuration composed of six traction wheels. This system showed to be effective in previous experiments in the sewer. The new traction and suspension systems are described below.

Traction System

The traction system is composed of two sprockets and a worm gear set. This mechanism allows to transmit the power from the motor to the wheel and also works as a speed reducer. The use of a worm gear set allows the transmission of movement in perpendicular directions. With this configuration it is possible to have the motor aligned with the wheel, reducing the overall size of the system. The gears are enclosed in a gearbox and the motor is connected to the gearbox by two aluminium supports. The traction system can be seen in Figure 2.6.

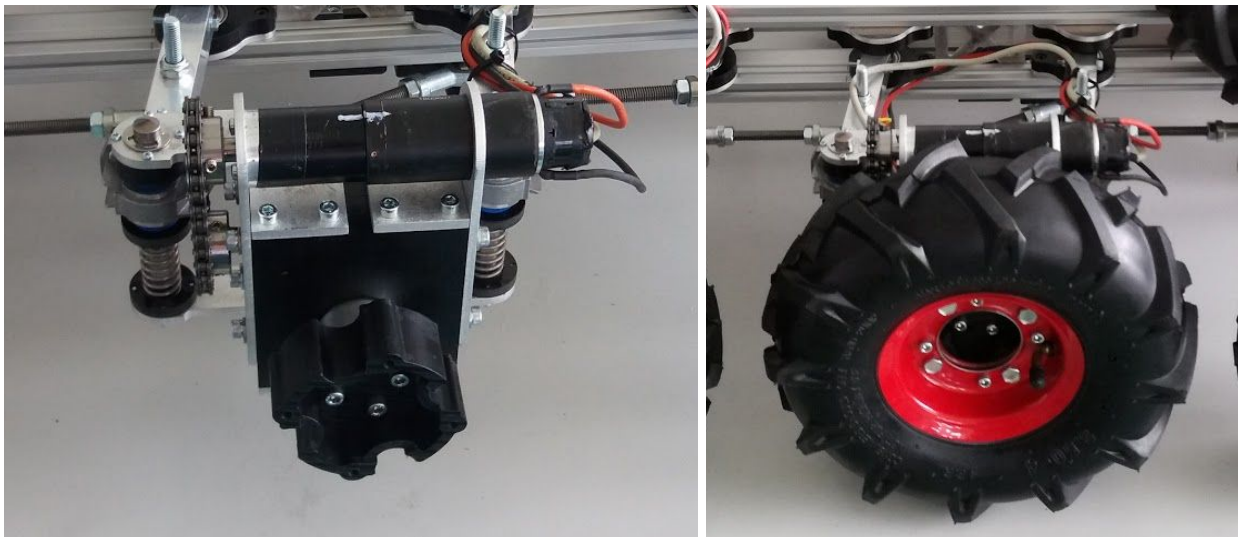

Figure 2.6: Left: traction system without wheel; Right: traction system with wheel.

As expected, the six wheel traction system configuration worked very well (as already experienced with the Phase I prototype). In order to ensure a good traction when moving both forward and backward, the four wheel in the front of the robot have the tyre´s treads in one direction and the back wheels have the treads in the other direction. This grants a good grip in both directions. A chain protection was added to each gearbox (see Figure 2.7). This protection was made in order to maintain the chain free of dirt.

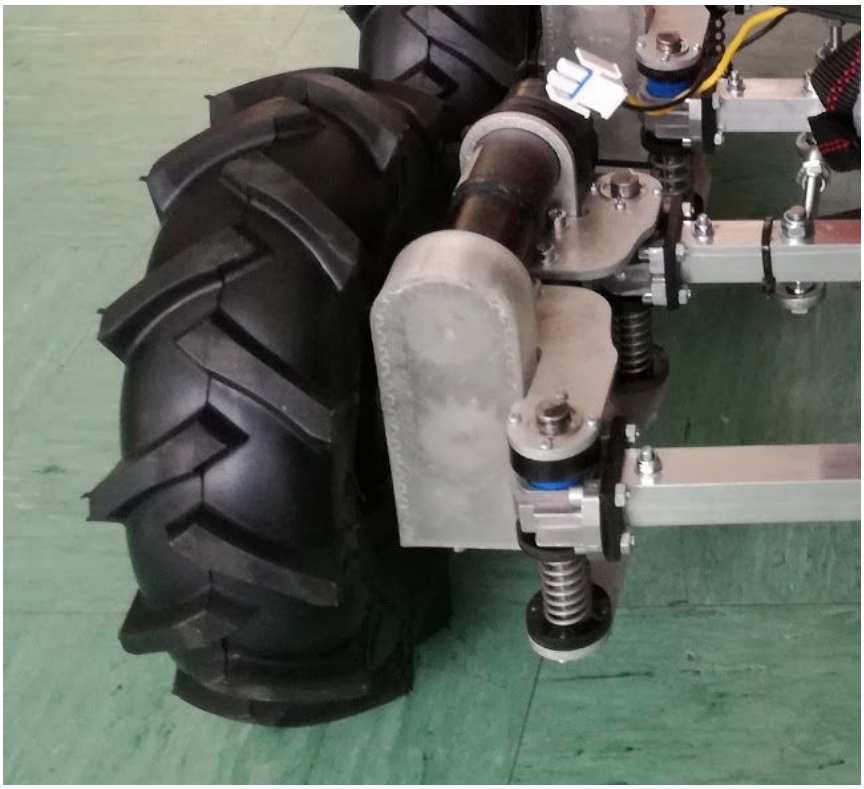

Figure 2.7: Chain protection (translucid plastic part).

Suspension

Due to the width adjustment mechanism, the introduction of a suspension mechanism became more difficult. Since the arms that connect the wheels to the frame are subjected to an angular variation the use of a traditional (car suspension) suspension was not possible. Instead, linear bearings were connected to the arms and then joined to the gearbox. Each gearbox has two rods that allow the linear bearing movement. Springs were coupled to the rods to limit the wheel displacement and to create a suspension mechanism. The suspension turned out to be very effective to overcome obstacles while maintaining the traction (wheels touching the ground). In Figure 2.8 the suspension working mechanism can be seen. Different spring rates were used in different wheels. The center wheels have springs with higher spring rates than the outer ones. The higher rigidity of the suspension of the center wheel creates pivoting axle through these wheel allowing the robot to turn easily. If all wheels had the same spring rate rotation would become more difficult. Figure 2.9 shows the robot using the suspension to overcome an obstacle.

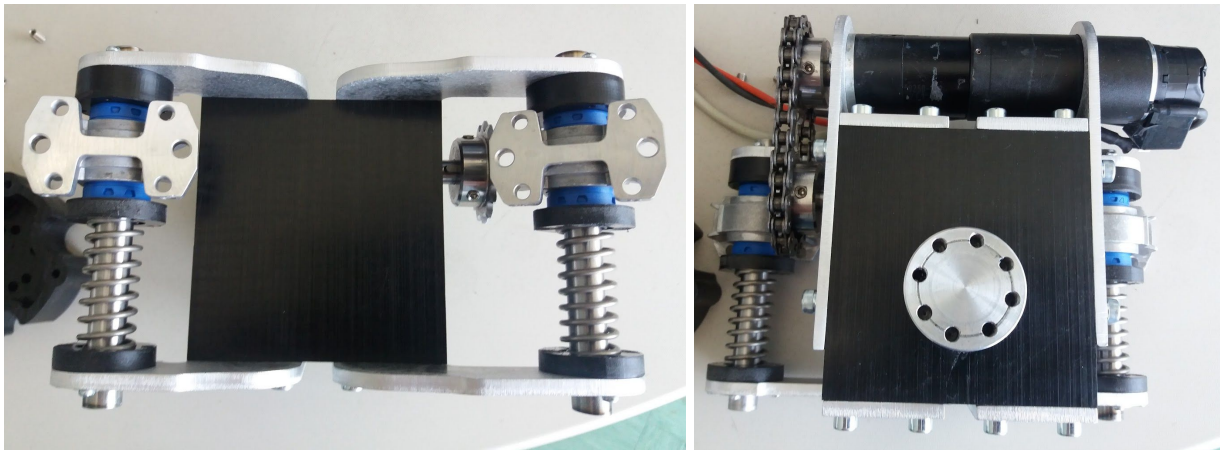

Figure 2.8: Two different views of the suspension mechanism. Left: Internal view. Right External view.

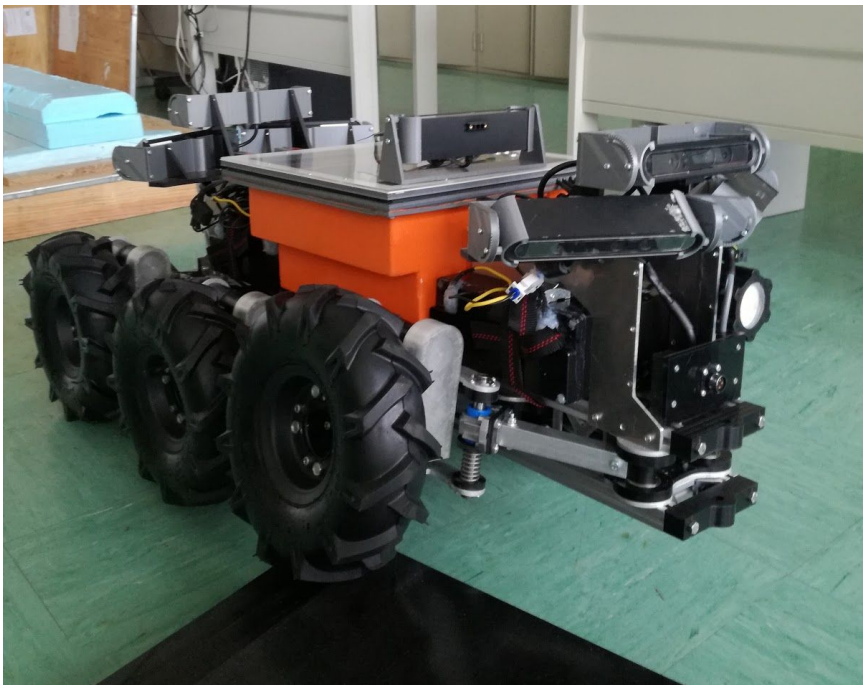

Figure 2.9: Suspension solicitation over an obstacle.

2.2.4. Cameras' Disposition and Supports

After some testing the following cameras and configuration are used in the SIAR:

1. Four Astra S cameras (two front and two rear) tilted 45 degrees (Figure 2.10 left) 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 (Figure 2.10 right) is possible to have a combined 110º field of view 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 supports maintain the cameras set in the correct position. The supports grant the correct horizontal positioning of the cameras (See Figure 2.11). The supports also have a scale with steps of five degree to allow the correct vertical positioning. The central camera, pointing to the ceiling also has a support, although, this one is a fixed support without angular regulation. Figure 2.12 shows the position of the 7 cameras on the robot.

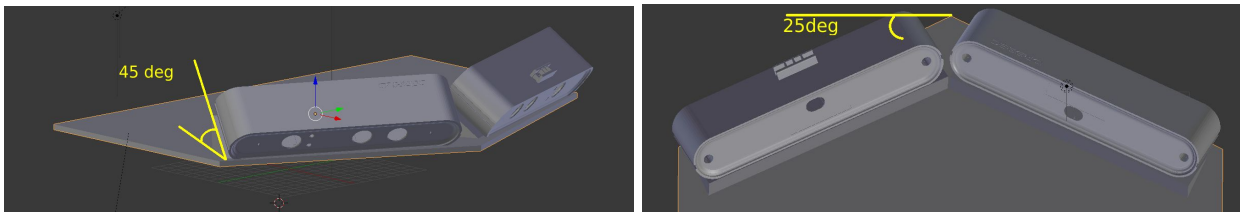

Figure 2.10. Left: tilt angles of the sensors. Right: top view of the sensing system, the yaw with respect to the horizontal of the robot is marked.

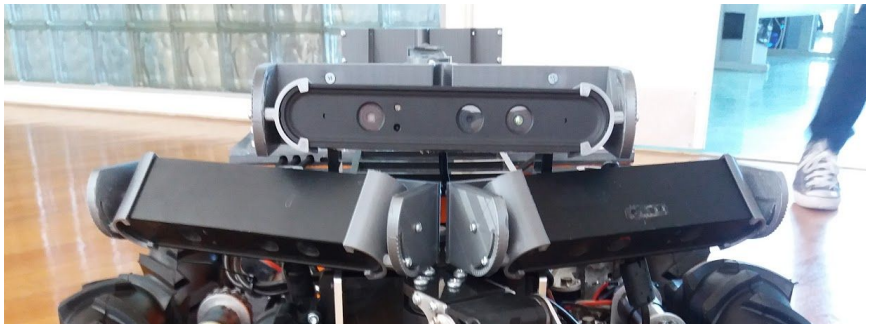

Figure 2.11: Back and Frontal cameras set.

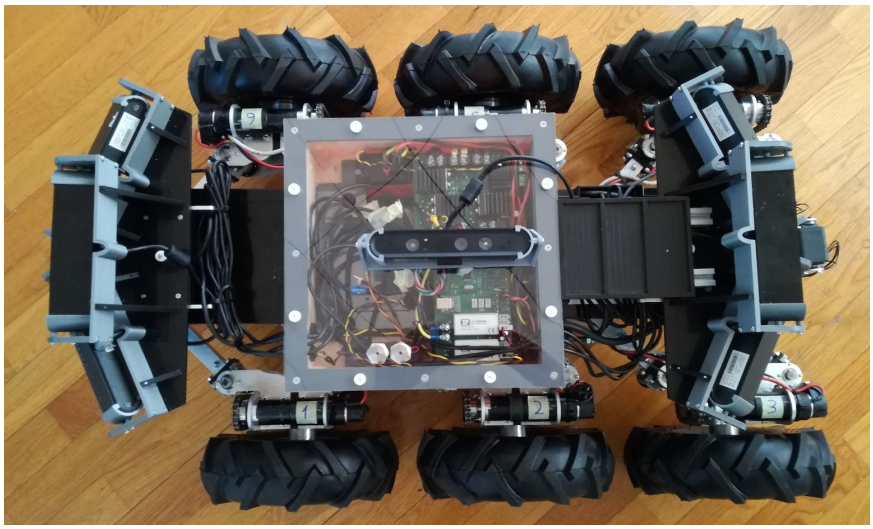

Figure 2.12: Top view of the seven cameras configuration.

2.2.5. Robotic Arm

The main goal of the robotic arm is the deployment and collection of the wireless repeaters in the sewer. The robotic arm has five degrees of freedom and it is actuated with the use of five Herkulex servos with different torques. The arm is composed of three segments. The claw is attached to the end of the third segment and it has the freedom to rotate. A new claw was designed in order to accommodate a camera. The camera is used to detect the handle of the wireless repeater to facilitate its deployment/collection. Around the camera there are three LEDs to illuminate specific points. When the arm is completely extended, it is able to reach a distance of 505 mm. Figure 2.13 shows the robotic arm with the new end effector.

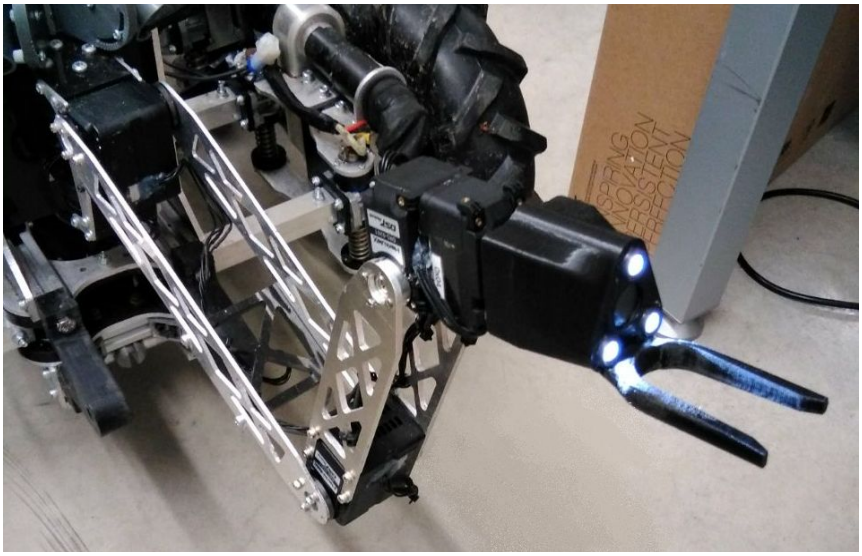

Figure 2.13. SIAR robotic arm with the new end effector.

Attention was paid to the robotic arm position in order to avoid interferences with the cameras in the rest position. The robotic arm is fixed at the rear of the robot. The repeaters are initially grouped on the top of the robot, behind the rear cameras set. The robotic arm has the ability to collect one repeater at the time and deploy it on the ground. The inverse is also valid, this is, the robotic arm has also the ability to collect the repeaters. Two stops were attached to the frame, one on each side of the robot to protect the robotic arm. The stops are connected by a pin that prevent the arm to move from the rest position.

A driver and a class have been written to control the arm at a high level of ROS. The arm driver could 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 impossible configuration. In the arm class, 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.

2.2.6. Isolation Solutions

In the sewer environment the robot is subjected to dirt and liquids. In order to protect the electronic components precautions were taken. All electrical components are inserted in a fiberglass box located at the center of the robot (see Figure 2.14). Tests were made to the box to ensure the it is waterproof. The box has a cover made of transparent polycarbonate. On the cover there are two IP67 buttons to connect the motors and the electronics. A water foam sealer was applied between the box and the cover to prevent water to enter the box. The cables from the traction motors, traction motors encoders, linear motor and the potentiometer enter the box, from the bottom, using IP67 cable glands. On the front and back faces of the boxes there are cable glands for the batteries, lights and the wireless module cables. On the joint between the box and the cover there are entrances for the USB cables from the Orbbec cameras. The geometry of the box was designed to minimize the volume of the robot and to allow the suspension movement without interferences with the box. All components connected to the box have IP67 or IP68 connectors.

In order to protect the robotic arm from liquids, silicon was applied to the servo-motor terminals. Polycarbonate protections were made to protect the lenses and microphones of the orbbec cameras. The terminals of the batteries were also isolated with silicon.

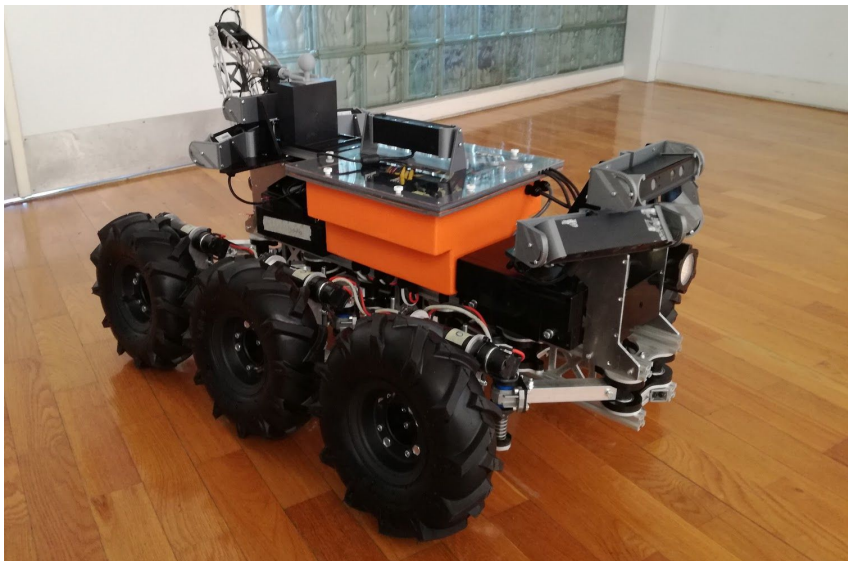

Figure 2.14: Orange electronic fiberglass box at the centre of the robot.

2.3. Robot Architecture

The system is composed by the following components:

- **NUC i7**. It is a i7 based processor that gets the information of the sensors inside the robot. It can communicate with the Control Center through a 2.4GHz link to send sensor information and receive commands.
- **Four Astra S cameras and three Astra cameras.** The cameras are the main sensor for navigation, localization and structure analysis.
- **SIAR Motor control board.** Is 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 add-in 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) but lower range (up to 10 Km with Line of Sight (LoS)) and robustness. This link will offer 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 sub-section describes the power and communication architectures between all the electronic device components.

2.3.1. Electronic Power Architecture

The robot platform is powered by two LiFePO4 12V 20AH battery packs. One battery pack delivers power to the motor drivers and the other provides energy to all other electronics (e.g. computer, electronics and sensors). Each of these packs, depicted in Figure 2.15, has the following specifications:

- Voltage: 12 V (12.8 V nominal)
- Capacity: 20 AH
- Charging Voltage: 14.6 V
- Charging Current: < 3 A
- Max Continuous Discharging Amperage: 20 A
- Maximum Discharging Current: 34 A
- Lifecycle of the whole pack: >85% capacity after 1000 cycles.
- Lifecycle of single cell: >85% capacity after 1500 cycles, >70% capacity after 3000 cycles. (<1C discharge rate and <1C charge rate).
- Low voltage (minimal): 9.5V
- Includes an embedded battery management system (BMS) circuit.

There is a total of approximately 0,25 KWh of installed power that provides an autonomy of 4 hours of operation. Additionally a power charger is able to charge the battery system in less than 4 hours. This Power Charger is able to power the robot electronics while it is in the charge mode. The batteries are installed in a way that can be easily accessed and replaced by another charged pack to continue with the operation.

One external battery charger and an external power supply can be connected to the robot allowing to power all robot components while charging both SIAR batteries. 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 2.16 depicts the onboard power architecture.

Figure 2.15. 12V 20AH LiFePO4 battery pack

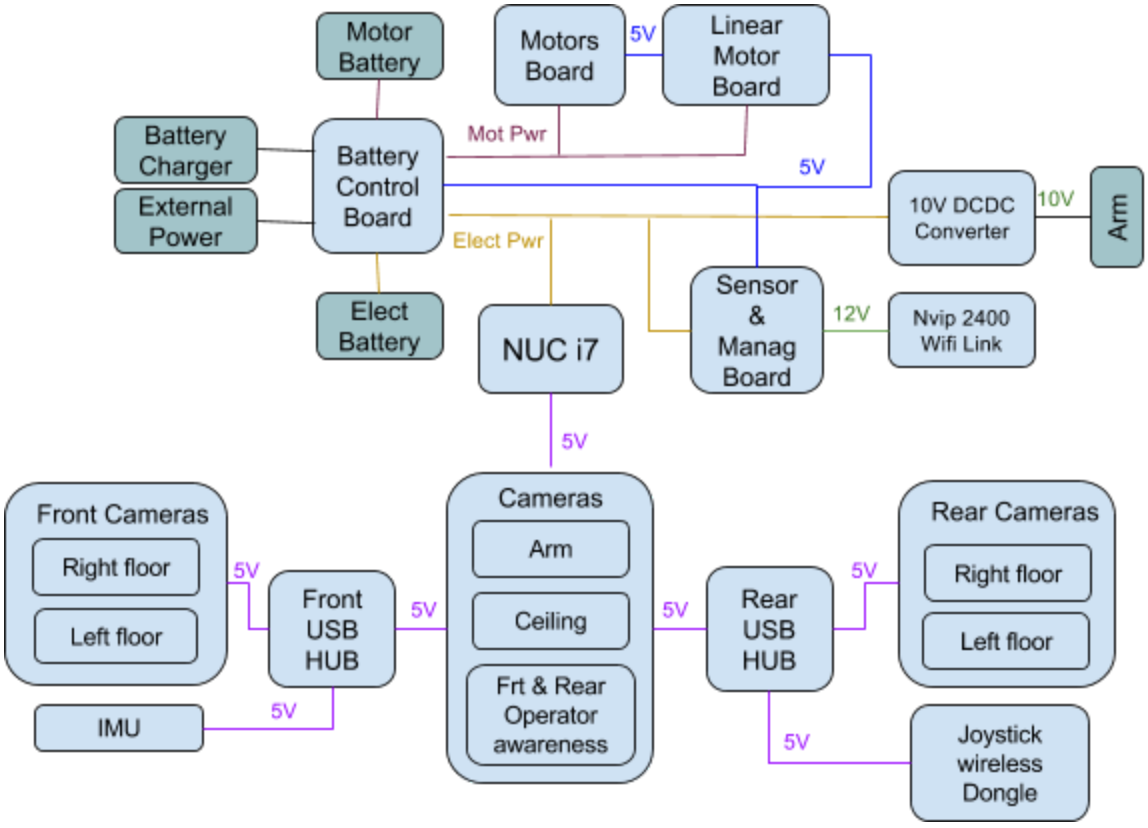

Figure 2.16. OnBoard Power Architecture

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 20Ah	Traction wheels motors	2,5	30
240W	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 20Ah	Sensor&Management Board	0,2	2.4
240W	Motor Board	0,15	1.8
	7x Orbbec Astra Camera	2,6	18,2
	Inertial Sensor	0,05	0,25
	2x Led Light Projector	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 3,5 hours.

Energy consumption measuring calculations

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 batteries, the remaining operation time is given by the battery that has a lower value. The Sensor&Management board calculates all the data for each battery.

The Sensor&Management board measures the actual voltage and the instantaneous current of each battery. This current measurement is performed by the use of a “shunt” resistor of 0.025 Ohms for the electronic battery and a 0.005 Ohms resistor for the motor battery. This “shunt” connects between the (-)

of the battery and the GND of all the onboard electronics. For the motor battery a maximum of 0.15 V (30 A) drop in the “shunt” resistor is expected and for the the electronic battery a maximum of 0.2 V (8 A) drop in the “shunt” resistor is expected. The current flowing through the resistor will be proportional to the voltage drop in that resistor. The microcontroller inside the Sensor&Management Board measures that voltage and calculate the instantaneous current. Then it integrates the instantaneous current over time and uses the integrated current to calculate the discharge level.

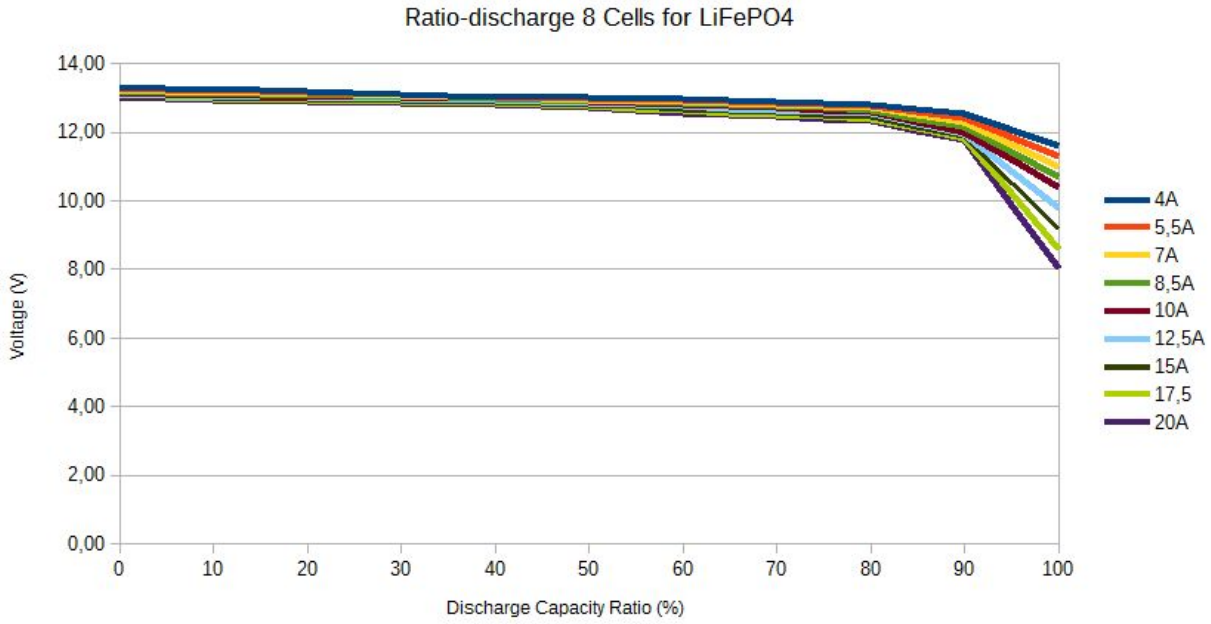

Figure 2.17: LiFePO4 discharge curves.

Each curve in Figure 2.17 represents an integrated current overtime *vs* actual voltage to determine the discharge level of the battery. LiFePO4 batteries, as it can be seen in the graphics, for a constant current will drop almost linear until they reach 80% of discharge, then they will start to drop faster. The voltage between two consecutive curves will have small changes, this fact allow to limit the number of curves that the microcontrollers have to process.

The curves are then represented in nine lookup tables, each one for a specific current. The 4 A curve is used to calculate the level of discharge for the integrated curves between 0 A and 4 A.

Each lookup table, for a constant current, will represent the actual voltage for a specific discharge percentage, in 5% steps.

The integrated current is used to determine the current curve lookup table and the actual voltage will give the position on the lookup table from where we can derive the discharge percentage of the battery.

Given the discharge percentage of each battery, it is possible to calculate the remaining time of operation (RTO), using the battery capacity (BC), the level of discharge (L) and the integrated current over time (C). The following equation is being used:

RTO = (BC/C) x 60(minutes) x (100-L) / 100

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 have been introduced to 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 2.18 shows the adopted electronic architecture for the Phase II solution.

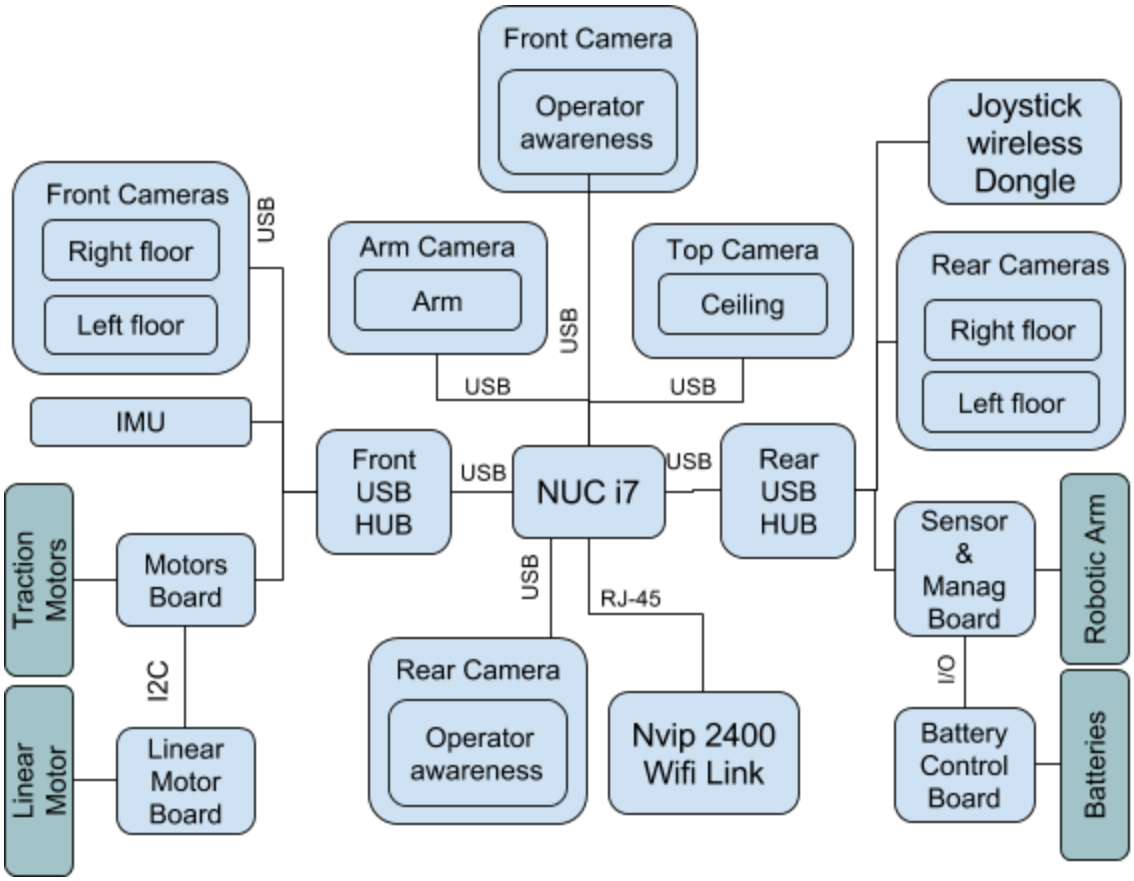

Figure 2.18: Electronics' architecture

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 PIC18F6527 microcontroller to control and manage all the communication between the NUC and the Slave Motor Driver Controllers. There are 7 slave motor control drivers: 6 that control the locomotion wheels and one motor that controls the width of the robot. The Slave Controllers Motor Driver Controllers use a PIC18F2431 microcontroller to provide all the necessary control signals to the motor drivers. The overall architecture of the Motor Controller Board is depicted in Figure 2.19. The Linear Motor Control board is an add in 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.

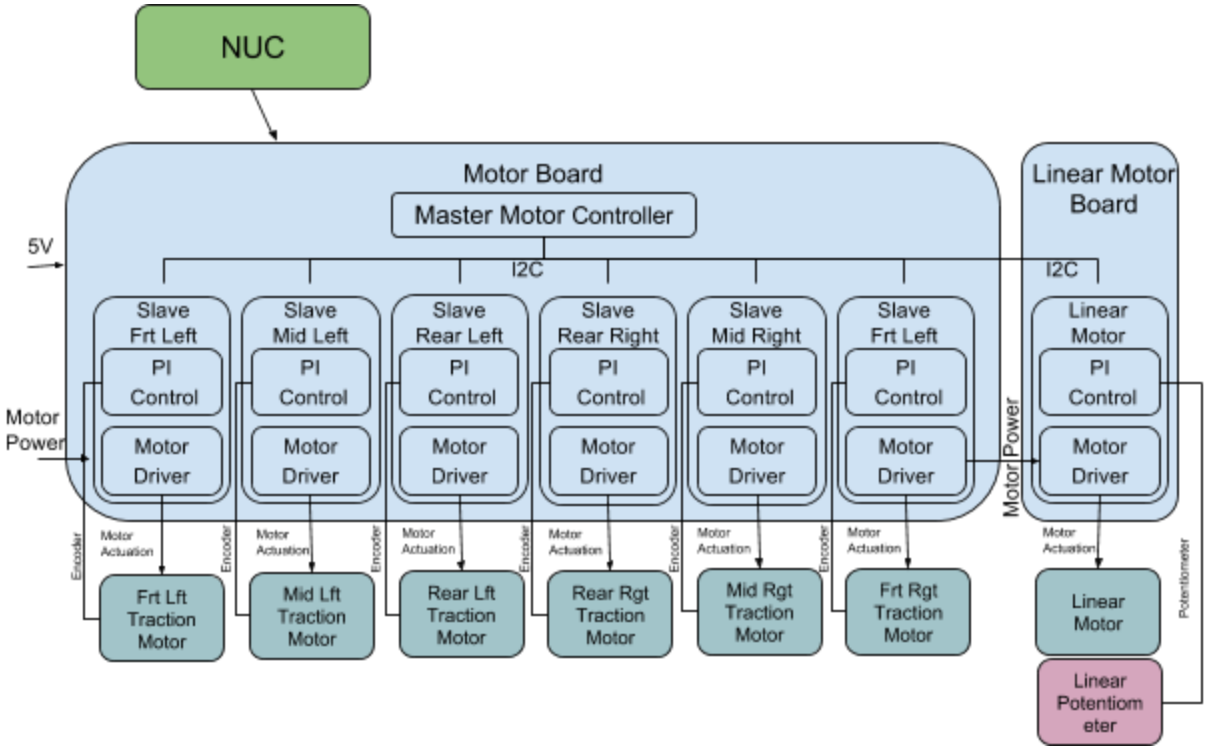

Figure 2.19: Motor Board architecture

The Master Motor Controller will:

- run low-level control loops to check for critical changes in the motor system that can affect the robot operation;

- provide 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;
- check the proper operation of each Slave Controller through exchange of status information;
- monitor the drivers' power supply;
- exchange information with the NUC;
- enable/disable the motor drivers;
- enable/disable the QuickStop control of the motor drivers.

The Slaves PI Traction Motor Controllers will:

- 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.

The Slaves PI Platform Width Motor Controllers will:

- receive width and velocity references from the Master Motor Controller;
- periodically read linear potentiometer value;
- send actual width 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 position and velocity acceleration/deceleration profile;
- remove the dead zone of the motor;
- limit 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. 2.20.

The Sensor&Management Controller uses a PIC18F6527 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 battery;
- 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.

In the future this board will be responsible for connecting a set of sensors and actuators that will be used in the project for environmental sensing. 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 change from batteries power to external power and charge.

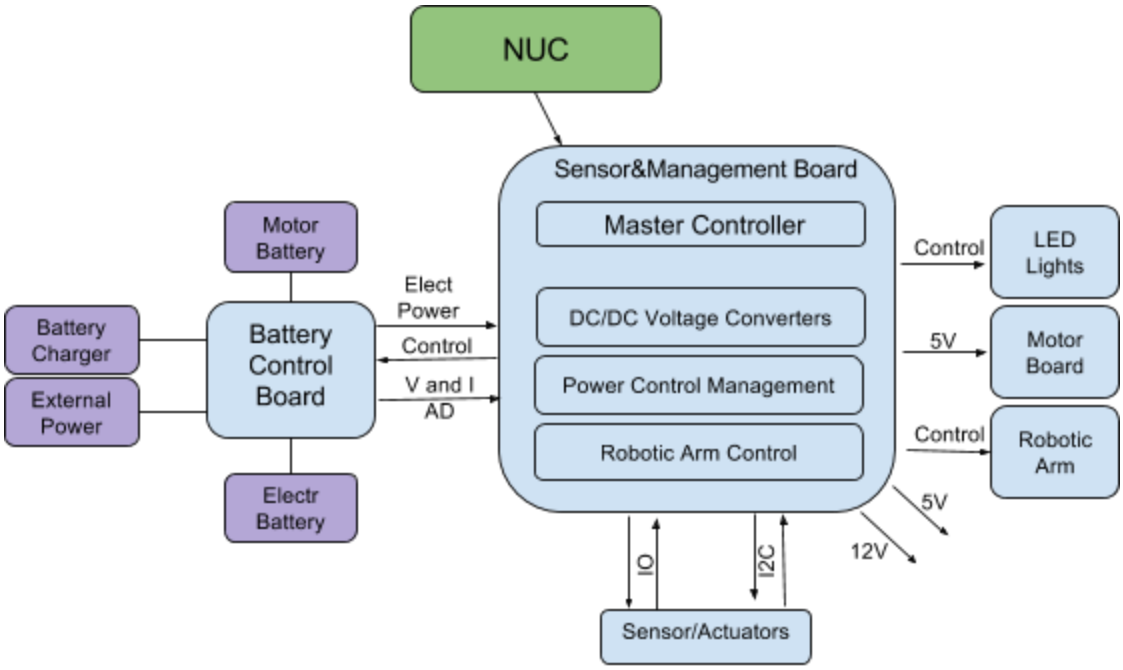

Figure 2.20. Sensor&Management Board architecture.

2.5. Improved Kinematics

The new SIAR robot was tested in the Barcelona sewers. These tests allowed to test the new introduced features in the real environment.

The first test was the deployment of the robot inside the sewer gallery (see Figure 2.21).

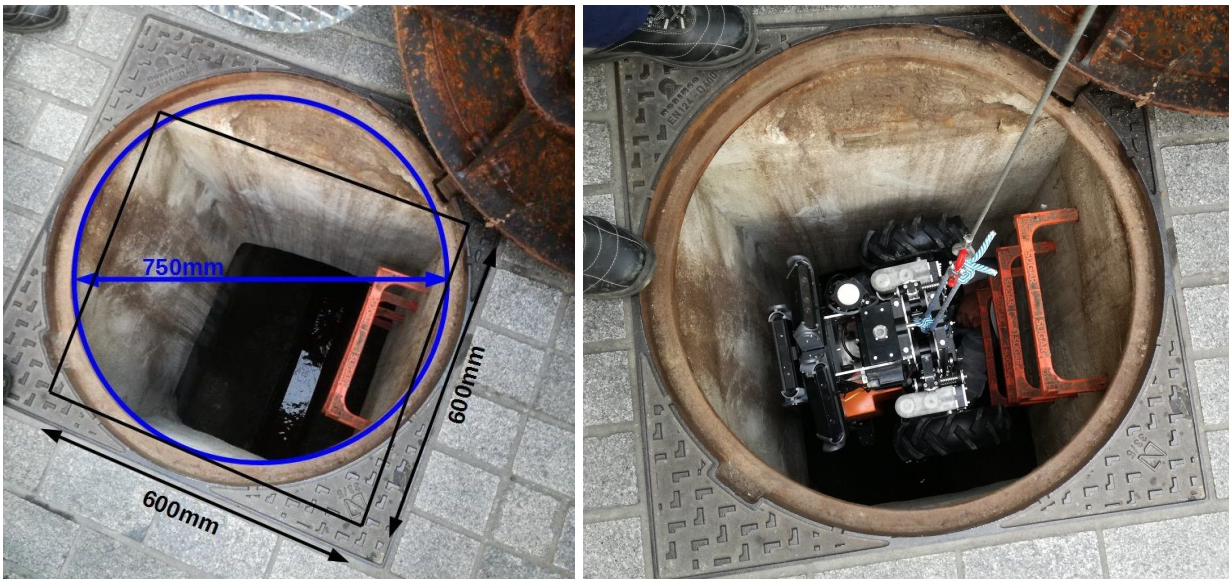

Figure 2.21. SIAR robot inside the manhole.

To enter the robot has to be able to go through a manhole that access the sewer gallery from the top. The manhole consists of a vertical hole with 75 cm of diameter. The interior has a squared section of 60 cm x 60 cm, with stair steps on one side. To pass the robot has to be with its minimum width configuration (50 cm width configuration). An electric winch is used to vertically deploy the robot inside the sewer. This is a two operators job. One on the top controlling the electric winch and aligning the wheels of the robot with the stairs and another operator on the bottom to align the robot wheels with the gutter.

The result was positive. The robot dimensions allow deployment inside the sewer without the need of removing cameras or other devices. To avoid hitting with the top cameras against the manhole walls, a top camera protection should be studied and created. This cover should be only used while passing through the manhole and removed after it.

The second test was to run inside the galleries over the gutter. Tests with different widths were performed. In the test area it is possible to find several types of galleries with different widths, gutters, forks and debris. In one of the areas the gallery is substituted by a wide pipe without a gutter, where the dirt water flows freely inside of it. Figures 2.22 to 2.24 show some of the described gallery areas.

Figure 2.22. Left: pipe gallery type. Right: transition from one gallery type to another gallery type.

Figure 2.23. Left: transition from one gallery type to another gallery type. Right: gallery fork.

Figure 2.24. Left: gallery debris. Right: pipe crossing the gallery walls

The result was very positive. It was easy to adjust the width of the robot to comply with the gallery and gutter width allowing to travel through the sewers at maximum robot velocity. Figure 2.25 show a gallery where the robot has to reduce its width to be able to run inside. The change of the width did not affect the normal movement of the robot. The change of the width in movement should be prefered in relation to a change when the robot is stopped. The movement reduces the lateral friction of the wheels on the sewer floor, reducing the required force to change the robot width.

Figure 2.25. Robot running over the gutter inside the gallery

The third test was the rotation over the gutter. The capability of the robot to make a full rotation over the gutter was tested. Figure 2.26 shows a 180º rotation over the gutter.

Figure 2.26. 180º rotation sequence over the gutter.

The result was very positive. To make a full rotation the robot should be set to its maximum width and the center of gravity of the robot should always be on the middle of the gutter.

The final test was the capability of the robot to cross or change direction in forks. Figure 2.27 shows the selected fork test zone.

Figure 2.27. Forks test zone.

In this set of tests the robot had to pass (Figure 2.28), turn left (Figure 2.29) or turn right in a fork, from the main channel to a secondary channel and from secondary channels to the main channel.

Figure 2.28. Continue in front

The results were again very positive. Several configurations were tested to cross the forks, like maximum width, change of center of gravity and minimum width to the fork central vertice. The robot was able to successfully pass the fork test in several configurations.

Figure 2.29. Turning left on a fork.

The tests were made inside the sewer galleries. The sewer galleries are dirt water collectors that are fed by small pipes connected to the surrounding buildings. The small pipes enter the gallery by the walls, creating small dirt water falls from the walls (Figure 2.30). During the tests the robot had to pass under some of these dirt water falls, making the wheels, mechanical structure and top components dirty. After the end of the tests, already on the outside, the robot was washed with a pressurized water pump. After cleaning the robot it was possible to understand that no electronic component was affected by the water and all the isolated electronic components were dry.

Figure 2.30. Small dirt water falls from the walls.

3. Communications

One of the main requirements of the robot is the capability of receiving commands from the operator, who, at the same time, should receive images and other data from the robot [ECHORD++, 2014]. This section details the implemented communication system that provides the SIAR platform with continuous wireless connection with the base station, even without direct line of sight (LoS) and with a long distance between the robot and the base station (hundreds of meters or even kilometers). Initial field experiments with the proposed equipment are presented in order to prove the validity of the solution.

3.1. Overview

Wireless communications in the sewage system is an open issue that would simplify the robot setup. Most of the current solutions use tethers and cables, which limit the distances that the robot can explore. The implemented solution aims to solve this problem by means of autonomous deploying self-powered wireless repeaters when needed.

The design of a wireless communication system inside the sewer system involves a great challenge due to the harsh environmental conditions and its architecture. In fact, it is common to find 100% humidity condition. Another major issue is the presence of obstacles, such as walls, pipes or garbage which would not allow LoS transmission between the robot and the base.

3.2. Solution

This section describes the proposed solution for communicating the base station with the robot. The proposed system is based on the use of robust 2.4 GHz WiFi communication repeaters that can extend the communication range even in presence of non-line of sight.

3.2.1. Communication device

The communication block is equipped with a Microhard nVIP-2400 long range 2.4GHz router. This device has been tested indoors and proved to communicate up to 300 meters with line of sight. This link will provide a high bandwidth connection that will be used for robot commanding, video streaming and additional information transmission such as the 3D reconstruction of the environment.

This module has already been described in D28.1 (please refer to the section 3.3.1 of this deliverable). To sum up, real-time teleoperation, with image feedback, was fulfilled from distance up to 300 m with two repeaters between the base station and the robot. These experiments have been confirmed in real sewer scenarios (see Section 3.3).

Table 3.1 shows the items that compose a repeater, their consumption or battery capacity and its weight and size.

Item	Power (W) / Capacity (Wh)	Weight (g)	Size (mm)
LiPo Battery (3S)	1460mAh@11.1V (33.3Wh)	100	137x 43x 19
nVIP2400	6W (max)	100	88,9x63,5x19
Enclosure		50	70x20x70
Antenna		30	70x2x2
Total		280	

Table 3.1: Items included in the repeater

3.2.2. Automatic deployment of repeaters

By deploying the repeaters we can extend the operational range of the robot to very long distances. The deployment should be performed whenever the LoS is expected to be lost or the signal starts to degrade.

Attention was paid to the robotic arm position in order to avoid interferences with the cameras in the rest position. The robotic arm is fixed at the rear of the robot. The repeaters are initially grouped on the top of the robot, behind the rear cameras set. The robotic arm has the ability to collect one repeater at the time (Figure 3.1) and deploy it on the ground. The inverse is also valid, this is, the robotic arm also has the ability to collect the repeaters.

Figure 3.1: Left: Taking the wireless repeater from the robot top.
Right: Deployment of the wireless repeater on the sewer floor.

Tests have been performed to find out the best way to detect the repeater (see Figure 3.2). In order to improve detection, AR markers have been chosen for long distances, and reflective tape for short ones.

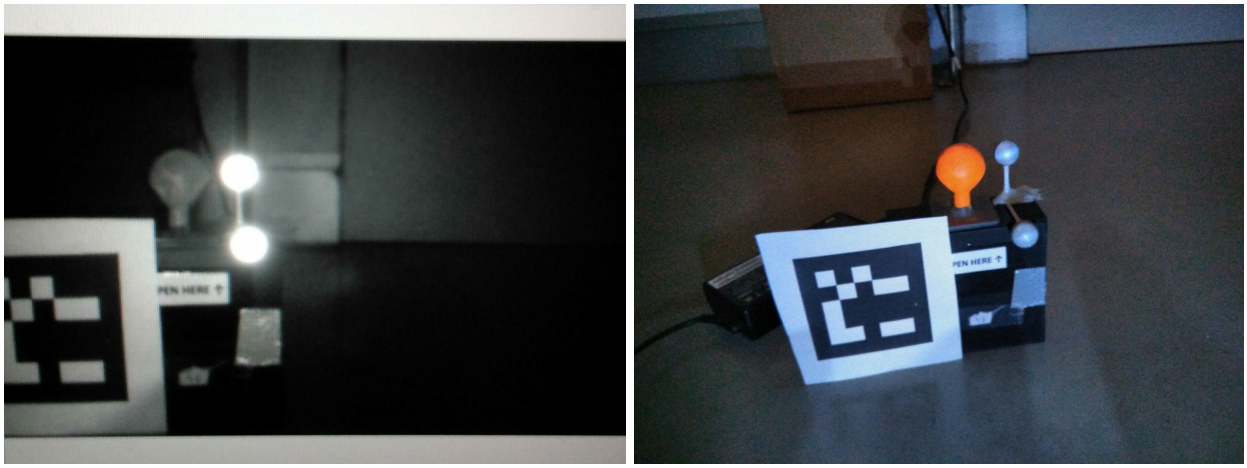

Figure 3.2: Test of the performance of three different types of markers.

To find the repeaters, the robots uses the three RGBD cameras on the back of the robot to search for the AR marker that is printed at both sides of the box. This will provide an estimation of the relative position between the robot and the repeater. The relative position of the repeater is estimated and the camera of the arm is moved to the proximity of the repeater's ball (see Figure 3.3). Then, the robotic arm camera detects the repeater ball, and uses a visual feedback loop to approach and grab the repeater.

Figure 3.3. The robot is picking up a repeater from the floor

To pick up and deploy a repeater, the robotic arm moves to a predefined position on the top of the robot. Then it uses the robotic arm camera to detect and approach one repeater. In this case, the AR marker is not used because it cannot be fully seen. In those cases, the use of reflective tape allows the markers to

be easily distinguishable in the scene when only the illumination provided by the arm and the robot are present.

Regarding the other movements used to deploy the repeater on the floor, to collect the repeater from the back of the robot and to leave the repeater back in the back, they will be predefined at a first stage. However, it could be necessary to add a perception procedure that could ensure that the deploy area on the floor is free of obstacles and also that the robot has been successfully collected from the bag or left back again.

3.3. Tests and experiments

In stage I, several tests were carried out at the basements of the University Pablo de Olavide (UPO) and in the sewers located at the two test environments: in the Passeig of Sant Joan and in the Mercat del Born. The results of these experiments were satisfactory: the communication was possible at each measured point going as far as roughly 500 m between the robot and the base station. Please refer to D28.1, Section 3.3 for more details about these experiments.

In this stage, experiments to validate the communication using several NVIP2400 repeaters in an Ad-Hoc network have been carried out in the Mercat del Born scenario. The tests are detailed as follows.

3.3.1 Experiments at the Mercat del Born sewers, July 2017

The wireless repeaters were deployed in the sewers in three manholes by fixing them to the stairs. The base station communication device was carried by the operator inside the sewer and deployed at a fork. This deployment does not involves extra costs, taking account that the operator should go inside for the robot deploying in the sewer galley.

Figure 3.4 represents the position of the repeaters and measuring points for this experiment. At each measurement point, a laptop with the robot's communication system was used to check for connectivity and to collect at least 20 measures of Round Trip Time (RTT). Then, the bandwidth between client and server was measured. Finally, Received Signal Strength Indicator (RSSI) and quality of the signal were available.

The tests allowed to make an estimation of the actual autonomy of each repeater. The Repeater 1 (IP:48) was the one that was more used from 12:20 to 14:00. In that period, it consumed 0.447 mAh. Thus, taking into account this data, a fully-charged battery can be used reliably (discharging it an 80%) for up to 4h (about 259 minutes).

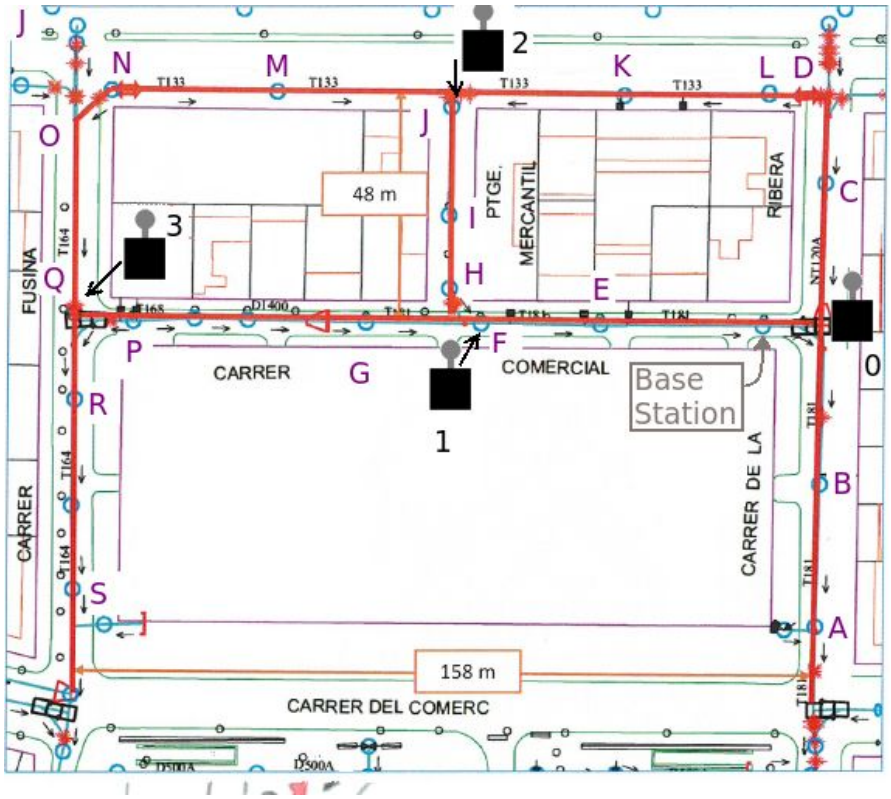

Figure 3.4: Disposition of the repeaters with manhole deployment

The results of the experiment are detailed in the tables below:

- Tests without repeaters in Carrer de la Ribera and Carrer Comercial. Table 3.2 shows the main characteristics of the link between the communication device of the robot and the base station without repeaters. This information includes the round trip time (RTT) mean and standard deviation. It also shows the measured Received Signal Strength Indicator (RSSI) and the measured bandwidth of the link.
- Tests with one repeater in Carrer Comercial. The results are shown in Table 3.3.
- Tets with two repeaters in Passeig Picasso. The results are shown in Table 3.4.
- Tests with two repeaters in Passeig Picasso. The results are shown in Table 3.5.

		A	B	C	D	E	F
Mean RTT (s)		1.429	1.812	1.987	4.580	2.030	2.376
Std Dev RTT (s)		0.4322	0.962	0.709	3.347	0.994	1.079
Bandwidth(Mbit/s)		14.2	15	12.8	10.3	14.7	10.9
Repeater 0 (base station) IP: 192.168.168.45 MAC: 00:0F:92:FA:A9:3F	RSSI (dBm)	-38	-43	-49	-74	-39	-61
	SNR (dB)	57	52	46	21	56	34
	Quality (%)	100	100	100	70	100	100

Table 3.2: Main characteristics of the communication without repeaters.

		G	H	I	J
Mean RTT (ms)		5.111	3.715	5.270	3.268
Std Dev RTT (ms)		2.680	2.181	2.421	1.258
Bandwidth(Mbit/s)		13.1	5.26	5.35	6.1
Repeater 0 (base station) IP: 192.168.168.45 MAC: 00:0F:92:FA:A9:3F	RSSI (dBm)	-89	-90	-89	-90
	SNR (dB)	6	5	6	5
	Quality (%)	20	16	20	16
Repeater 1 IP: 192.168.168.48 MAC: 00:0F:92:FA:3E:9A	RSSI	-36	-51	-61	-58
	SNR	59	44	34	37
	Quality (%)	100	100	100	100

Table 3.3: Main characteristics of the communication with one repeater.

		K	L	M	N	O
Mean RTT (ms)		7.693	4.580	5.295	8.387	11.353
Std Dev RTT (ms)		5.324	3.347	2.959	15.802	8.269
Bandwidth(Mbit/s)		2.57	2.18	3.37	3.28	3.2
Repeater 0 (base station) IP: 192.168.168.45 MAC: 00:0F:92:FA:A9:3F	RSSI (dBm)	-87	-83	-79	-76	-76
	SNR (dB)	8	12	15	19	19
	Quality (%)	26	40	53	63	63
Repeater 1 IP: 192.168.168.48 MAC: 00:0F:92:FA:3E:9A	RSSI	-92	-92	-93	-90	-90
	SNR	3	3	2	5	5
	Quality (%)	10	10	6	16	16
Repeater 2 IP: 192.168.168.49 MAC: 00:0F:92:FA:4C:AC	RSSI	-50	-56	-57	-43	-73
	SNR	45	39	38	52	22
	Quality (%)	100	100	100	100	73

Table 3.4: Main characteristics of the communication with two repeater.

		P	Q	R	S
Mean RTT (ms)		7.901	1.908	4.395	5.512
Std Dev RTT (ms)		4.540	0.9445	3.806	3.008
Bandwidth(Mbit/s)		9.96	1.2	2.41	2.29
Repeater 0 (base station) IP: 192.168.168.45 MAC: 00:0F:92:FA:A9:3F	RSSI (dBm)	-62	-75	-93	-94
	SNR (dB)	33	82	2	1
	Quality (%)	100	66	6	3
Repeater 1 IP: 192.168.168.48 MAC: 00:0F:92:FA:3E:9A	RSSI	-57	-68	-91	-90
	SNR	38	27	4	5
	Quality (%)	100	90	10	16
Repeater 2 IP: 192.168.168.49 MAC: 00:0F:92:FA:4C:AC	RSSI	-88	-90	-89	-91
	SNR	4	5	6	4
	Quality (%)	23	16	20	13
Repeater 3 IP: 192.168.168.46 MAC: 00:0F:92:FA:3E:9B	RSSI	-4	5	-47	-71
	SNR	91	100	48	24
	Quality (%)	100	100	100	80

Table 3.5: Main characteristics of the communication with three repeaters.

The results of this experiment were satisfactory to a great extent. First, total scenario coverture was achieved by the use of three manually deployed repeaters in the manholes. Please note that the repeaters were not directly deployed in the forks to achieve line of sight. Instead, we wanted to test a rapid manual deployment that can be done in the future without the need of descending through the manhole.

Second, the quality of the communication link was above 70% in all cases, despite of the lack of the line of sight in some cases. A minimum of 1Mb/s of bandwidth was always achieved, that is enough for the software to provide a real-time RGB and depth images of one camera at a 10Hz rate.

On September, 21st, this deployment mode of the repeaters was used to successfully tele-operate the robot in manual and assisted modes from the surface, during the whole track. The image flow provided to the operator was stable. This allowed an operator from BCASA to tele-operate the robot in semi-autonomous mode, from the base station, over the complete track. This was performed by an operator without any prior training.

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 System[1] (ROS) framework, using the ros-indigo distribution under Ubuntu 14.04.

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

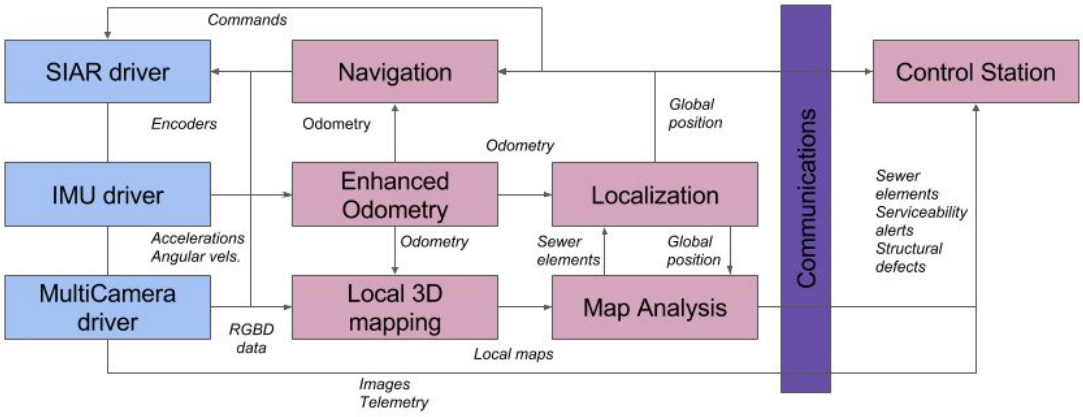

Figure 4.1. Software Architecture

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 a 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 a utility for generating launch files that sequentially activate each camera. Each camera can be configured with its own setup, i.e. changing the image rate, its resolution and many other parameters.

[1] http://ros.org

[2] http://wiki.ros.org/openni2_camera

[3] http://wiki.ros.org/openni2_launch

- Enhanced Odometry (Section 5.1). 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 precisely and robustly estimate a 6-DOF odometry. A good odometry estimation is necessary for obtaining precise 3D maps of the environment.

- Localization (Section 5.2). The precise odometry estimations are used by this module in order to estimate the global position of the platform. This block uses prior information of the sewer systems (localization of manholes, inlets, to name a few), 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 5.3). The SIAR robot will be commanded in three different operating modes. This module will also allow the operator to command actions to the SIAR which are not directly related to navigation, such as taking air or water measures and deploying repeaters.

- Local 3D mapping (Section 6.2). 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 (Sections 6.3, 6.4 and 6.5). 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.

- Control Station (Section 4.1): This module will send an alert to the operator 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 will also display the localization of the robot and the images from the onboard cameras in real-time.

- 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, we opted to bring up two independent ROS cores in both the robot and the base station and to design and implement new ROS modules that would act as bridge between the different ROS cores in the system.

4.1. SIAR Control Station

In this section the design of the information that will be available at the control station is detailed.

Figure 4.2 shows the main window of the control center. This window shows real-time images from the onboard cameras that are specifically placed for operator awareness purposes. The application will switch between the front and rear cameras according to the operation mode, i.e., it will show the rear camera when the reverse mode is on, or the front camera otherwise. Additionally, the system shows the approximate location of the robot over a map of the environment as well as the location of deployed repeaters and alarms regarding to structural defects or serviceability losses in the sewer system.

FIgure 4.2. Main control window. Left, real-time images gathered from the front camera. Right: the location of the robot (blue) and the manholes (green) and forks (red).

The current software, which includes the main operator window, has been successfully tested during the experiments carried out in the “Mercat del Born” area in May 22-23, 2017. In particular, during the first day of the experiments, an operator from BCASA was 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 50 m. Figure 4.3 displays a picture taken in the experiment.

Figure 4.3. An operator from BCASA safely guides the robot in semi-autonomous navigation mode from the outside of the sewers.

In the second screen, this map is presented larger, and each system will be selectable. When selected, additional information about each element is displayed (e.g., battery status, pictures and characteristics of the detected defects, etc.). The operator is allowed to change and update the information, as well as to create new alerts whenever required. This part of the system is still under development.

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. This section details the characteristics and compares it to the standard ROS middleware.

4.2.1. Standard ROS middleware

ROS provides the user with a middleware that could be used for connecting different machines. The main characteristics of the middleware are listed below.

- Great versatility of types of communication:
    - *data-centric* communication, in which publisher and subscribers of different data topics can be created.
    - *service-centric* communication, in which servers can provide clients with services.
    - *action-oriented* services[4].
- The transmission is carried out only through TCP when processes from different computers are connected.
- Acts as a nameserver, which is used to generate TCP connections. Once the different modules are connected, the communication does not need to pass through the nameserver node.
- Multiplatform. It can be used in all platforms where ROS can be installed.
- Provides a distributed parameter system for configuring the ROS nodes.

Therefore the ROS middleware tool is a well-known and reliable tool that could be used for communicating the robot and the base station. However, we opted not to use it for this purpose for the following reason:

- It only uses TCP for connecting the different computers. This turned out to be a critical issue when real-time data should be transmitted, as the reliability comes at the expense of retransmitting lost packages. Therefore, the delay of the obtained data cannot be controlled.

[4] http://wiki.ros.org/actionlib

4.2.2. Proposed communication middleware

Figure 4.3 shows the block diagram of the proposed middleware solution. In this solution, ROS middleware is employed for connecting the internal nodes for connecting 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 topic 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 should be executed in the robot computer whereas the client node should be executed at the computer of the base station. These nodes can be configured to transmit the desired topics of interest at different rates at the request of the operator.

Finally, it is important to mention the modules that are being used to transmit real-time RGBD images to the operator in real-time. 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 would not be able to transmit the information of all cameras at the same time. Therefore, a camera switch module has been designed. It allows the operator to select the camera or cameras of interest to be transmitted, to configure the quality of the images and to enable or disable the transmission of the depth images in real time.

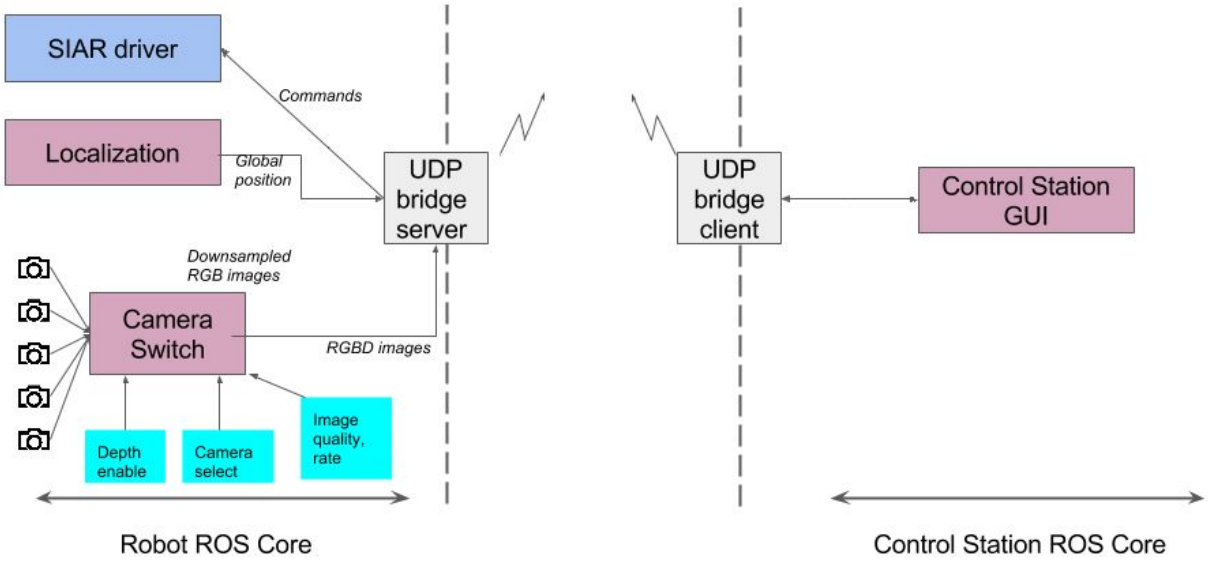

Figure 4.3. Proposed architecture of the middleware of the SIAR solution.

5. Localization and Navigation

The added value of a robot with respect to current systems is its capability of performing autonomously the inspection tasks. The basic functionalities required are localization and navigation. Localization is required to register the results of the inspection into the sewer network maps. It is also a requisite of the navigation module, which will allow the robot to move autonomously through the sewer in case it is needed. The following sections describe the designed modules, and, in some cases, preliminary results obtained.

5.1. Enhanced Odometry

Robot odometry is one of the main elements into the localization architecture. The more robust and accurate the odometry is, the better is the robot localization in general. Ground robots use to rely on wheel encoders and inertial units to estimate its odometry. While this is a good sensor combination with proven good results in many scenarios, the environmental constraints of sewers navigation forces us to include more sensing modalities in order to have an accurate robot odometry. Thus, humidity, water and waste significantly decrease the wheel grip, distorting the computed linear and angular velocities based on wheel encoders. Last, but not least, the robot will need to navigate into a full 3D environment, so the robot localization will also need full 3D odometry for 3D environment mapping and navigation.

The main sensors used by the SIAR odometry system are the RGB-D cameras mounted in the front and the rear of the robot, the 3D Inertial Measurement Unit (IMU) installed in the robot chassis and the wheel encoders. The main processing modules are presented in Figure 5.1.

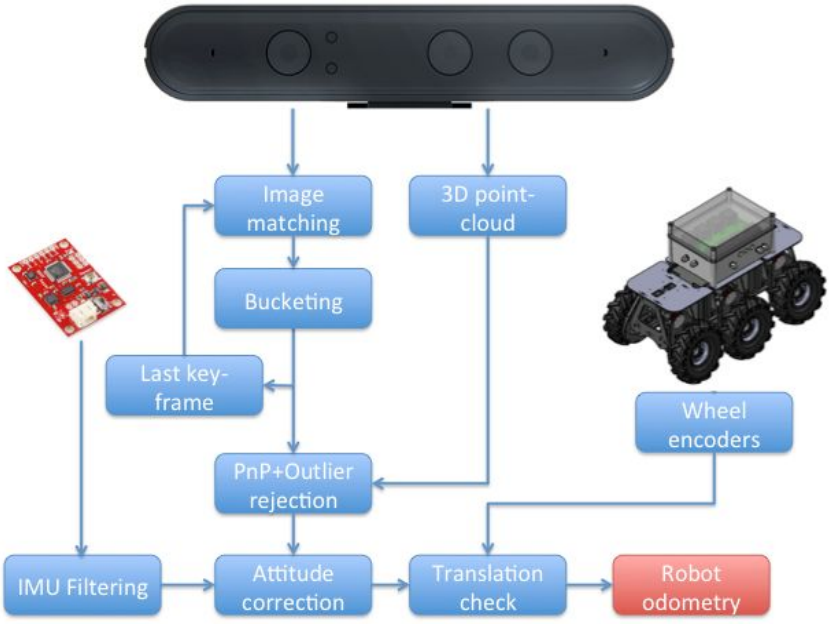

Figure 5.1: SIAR Robot odometry system

The RGB-D camera is used to extract the robot rotation and translation between frames, and the IMU is used to stabilize the robot roll and pitch that are fully observable from the IMU accelerometer. In addition, the estimation is double-checked with the velocity provided by the wheel encoders so that the system can face wrong estimations provided by the vision system.

The full odometry system was already presented with detail in Deliverable D28.1, please refer to this document for further details.

5.2. Localization

According to the requirements on [ECHORD++, 2014], one of the main functionalities required from SIAR is the registration of the monitored elements on the sewer network. This requires a localization system able to estimate the robot's position with respect to the previously existing map of the sewer system.

The proposed localization system is based on the integration of visual odometry, visual detection of manholes and Monte-Carlo localization. A brief description of the main functional blocks of the proposed approach is introduced below (see Figure 5.2). This system has been published as a part of the International Conference on Intelligent Robots and Systems, 2017 [Alejo et al., 2017].

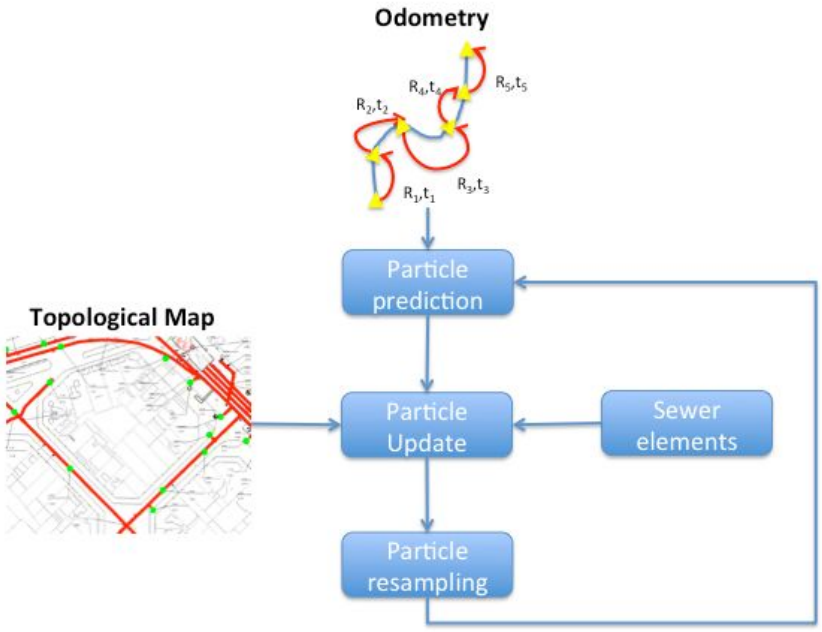

Figure 5.2. Topological map based localization approach. The detected sewer elements, like manholes, and a map of the sewer network including such elements are used to provide a global localization tracking within this map.

- **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 RBG-D based Odometry can be found.

- **Manhole detector**. This module will check whether the robot is under a manhole or not based on the depth images gathered by a camera pointed upwards (please refer to Section 2.4.1 for details on the disposition of the sensors). A machine learning approach is employed to perform this classification robustly and computationally efficient. It has been described in Section 4.2.3.

- **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 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. This utility is described in Section 5.2.2.

5.2.1. Graph-based Localization

As previously introduced, the approach is based on MonteCarlo Localization [Thrun et al., 2001], which makes use of a particle filter to represent the robot localization belief. In the proposed filter, each particle represents a hypothesis which consists of a 2D position with orientation. The hypotheses are validated (weighted) according to the position of each particle compared with a topological map obtained from GIS data. Figure 5.3 shows a detail of the GIS information used in the experiments to extract a graph that will be internally used by the localization module is extracted.

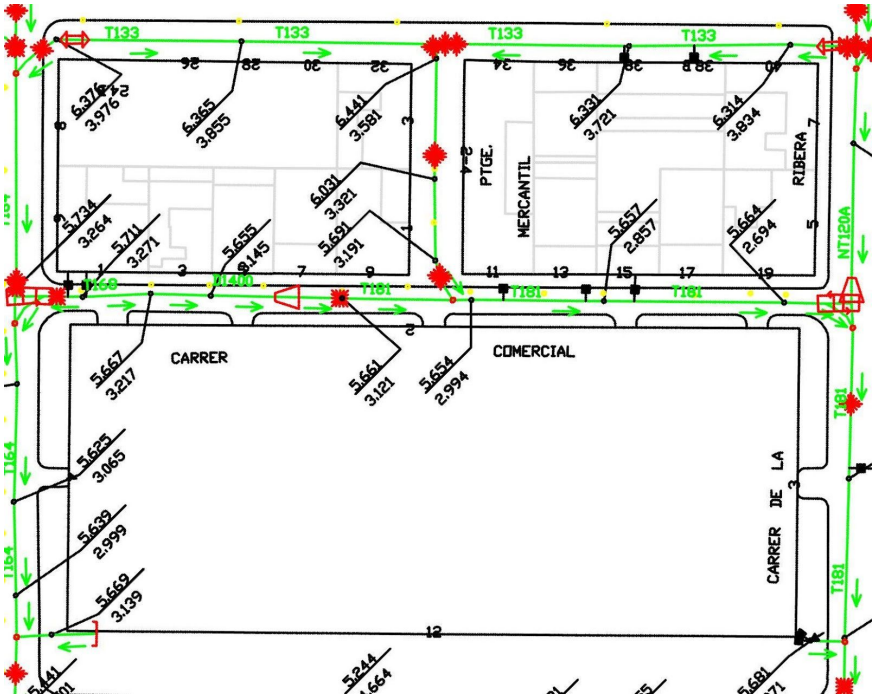

Figure 5.3. GIS information used as input for generating the internal graph in the localization module.

This internal graph contains manhole vertices, where a manhole detection can be performed; and fork vertices, in which whether the several sewers converge, or a the sewer starts to turn to another direction. These vertices are joined with edges that indicate traversable paths between them. Also, additional information regarding to section types, altitude of the sewer and manhole, to name a few, can be added.

Next, we describe the main steps of the localization system.

1. Initialization. The initial pose of the robot is set according to the position of the manhole where it is deployed and is orientated towards the next PoI of the inspection plan.

2. Prediction. The position of the particles is updated in this step according to the odometry measures.

3. Update. Once the new position of the particles is calculated in step 2, they are assigned a fitness value depending on how well they fit to the topological map. In case in which a manhole is detected, the fitness value depends on the distance to the closest manhole. Otherwise, the fitness will depend on the distance to the topological map.

4. Resampling. In the proposed system, the set of particles is periodically resampled once for each n updates. By periodically resampling the set of particles, we ensure that most of them are located according to the GIS information.

5.2.2. GIS information extractor utility

As specified in [ECHORD++, 2014], the robot platform should be capable of visiting up to 800 km of visitable sewers in Barcelona. To make it possible, it is convenient to extract the useful GIS information obtained from local agencies. This information can be used for planning missions and for localization purposes (see above). Furthermore, this is also necessary as the alerts generated by serviceability and structural issues should be geolocalized according to that GIS data.

For these two main reasons, we developed a simple utility that can read the DXF (Autocad) file provided by BCASA and generates a local graph in a desired area that can be loaded by the Localization module. Figure 5.4 represents a DXF input example and its associated sewer graph that can be used by the localization module.

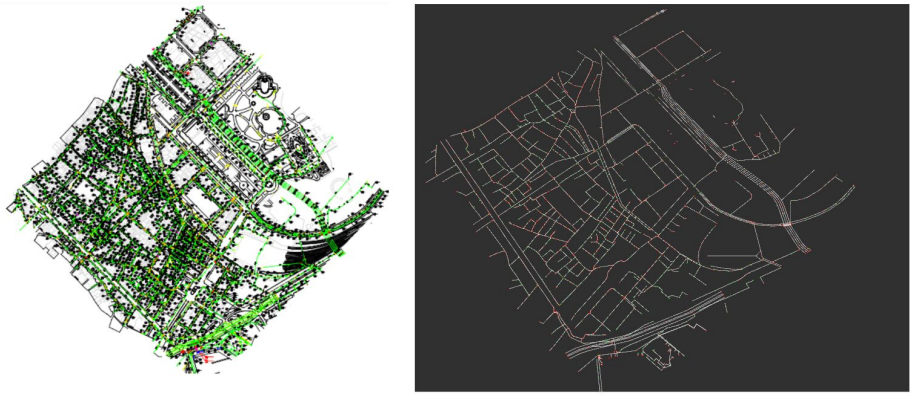

Figure 5.4. Left: The complete GIS information as given by BCASA. Right: The extracted graph that internally represents the geometry of the sewer system.

5.2.3. Field experiments in September, 20th, 2017

The proposed system has been proven extensively during Phase II. Please refer to D28.5, where we analyzed and presented the results obtained in two different experiments that were carried out in the 17th of January with very promising results. These experiments have been published in the paper [Alejo et al., 17] and presented in the IROS 2017, Vancouver.

In this section, we provide an additional result of localization with the data acquired in the last experiments of September, 20th. The system with the same configuration as in the experiments of January was executed with the new data, giving the results detailed in Figure 5.5. In this case, the experiment covered almost all the sewers that should be inspected in the final demo of Phase II, with the exception of a part of Carrer Ribera. This figure represents the weighted mean position of the particle cloud at each instant. The results are quite good in the majority of the experiment, with the exception of a small fraction that appears in the upper-left part of the figure. In this case, the particle cloud held more than one hypotheses and therefore the mean is not a good representation of the information in the cloud. Nevertheless, the system was able to properly localize the robot during all the experiment.

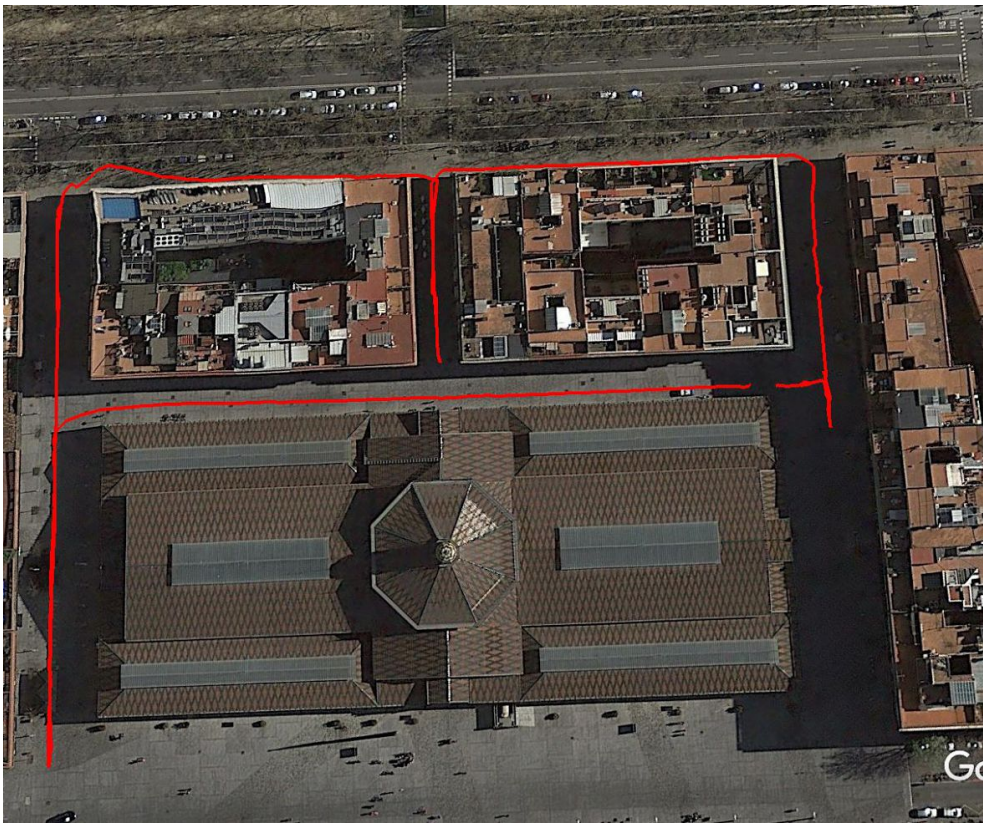

Figure 5.5. Localization results of the robot's position in the experiments of September, 20th.

5.3. Navigation

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 solving the navigation so that the operator can focus in the inspection task, even if some sections of the sewer are difficult to navigate. Thus, this module provides semi and fully autonomous navigation capabilities through the sewage system.

We envisage three main navigation modes that will be offered by the navigation module:

- Teleoperation: in this mode, the robot can be purely teleoperated from the base station, where the operator controls 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: this method has been developed in order to make the robot automatically navigate 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.

- Finally, a fully autonomous operating mode will be implemented. This operating mode will further reduce the operator's workload, as complete inspection plans could be loaded to the platform, that will execute it automatically. Moreover, it would be crucial in case where the communication was lost. In this way, the platform can safely perform an automatic return home procedure.

As indicated in D28.4, two main modes of operation are considered: assisted teleoperation and autonomous mode. In both modes, the robot should be 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 5.3.1) takes care of this part. The robot should be able to negotiate forks and bifurcations, which typically require the capability of performing maneuvers. This is explained in Section 5.3.2.

5.3.1. 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. To date we have empirically set a threshold of 7.5 cm as the maximum step that can be safely negotiated by the platform. 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. Figure 5.6 shows an altitude map obtained from an experiment in the sewer mockup of the University Pablo de Olavide.

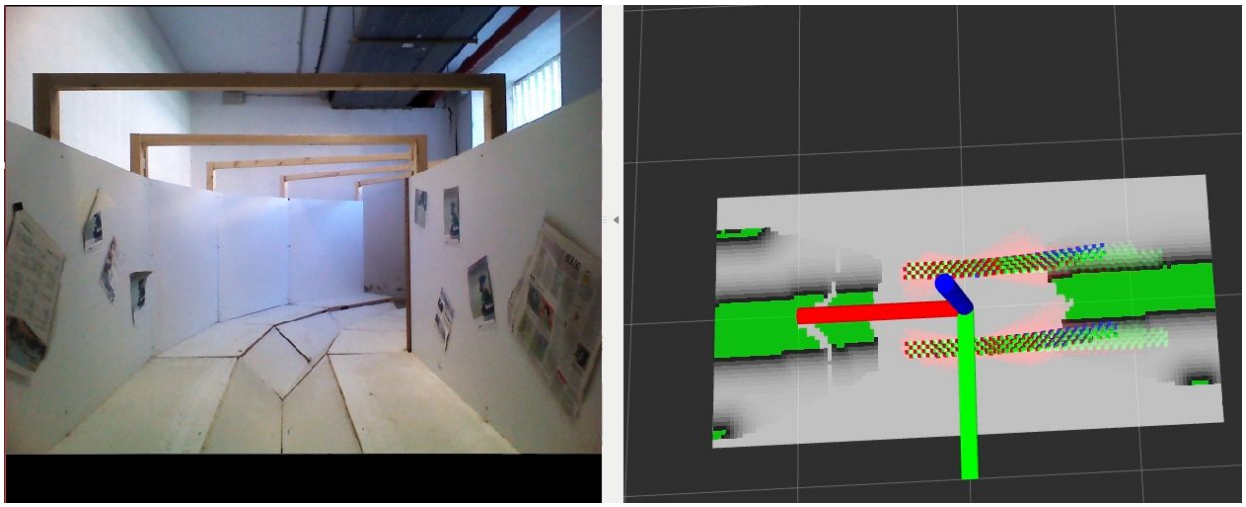

Figure 5.6. Left: picture obtained from the operator awareness camera. Right: associated traversability map, with untraversable zones in green (note the detected gutter in the middle) and checked trajectories: in red untraversable, in green traversable and best trajectory in blue.

This traversability map is then used for computing safe trajectories when the robot operates in assisted teleoperation 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 5.6 (right) depicts the 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 5.6 (right). This operation is also repeated at a frequency of 10 Hz.

The local navigation module has been successfully tested in the custom made mockup of the sewers at the University Pablo de Olavide and the real sewers of "Mercat del Born" in several experiments in May, 2017 and Sept. 2017. Figure 5.7 represents two snapshots of some of these experiments.

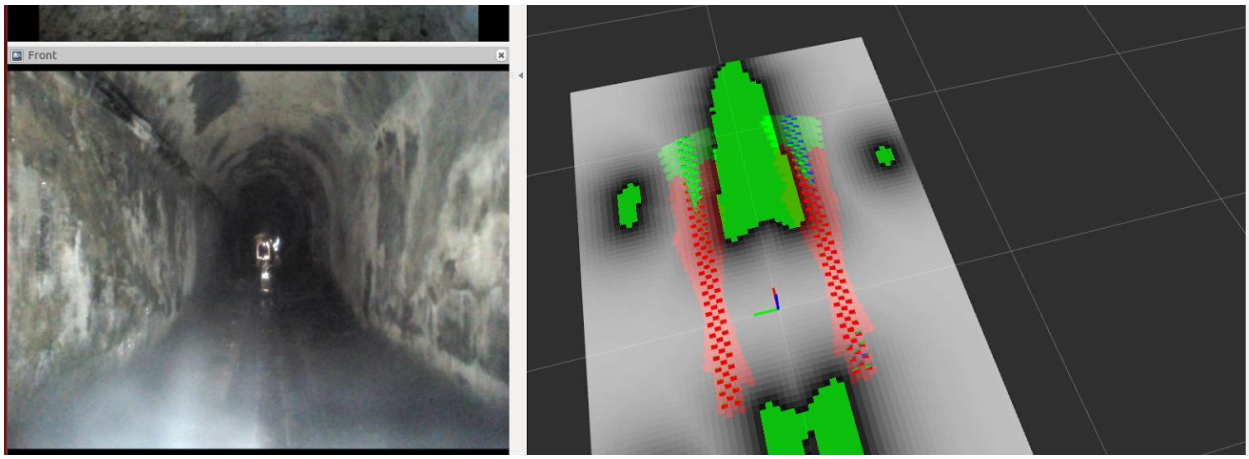

Figure 5.7. Left: Picture obtained from the operator awareness camera. Right: Associated altitude map and checked trajectories: in red untraversable, in green traversable and best trajectory in blue.

5.3.2. Local 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 bases its reliability in not letting any wheel to enter the gutter.

The aforementioned automatic control of the position of the robot with respect to the gutter can be modified in a way that lets the robot not only cross the gutter in presence of forks but also change the direction whenever required.

To this end, we modified the concept of traversable areas including the following different traversability definitions as a function of the action being performed by the robot:

- **Normal mode:** in normal mode, no part of the wheel is allowed to enter the gutter. Therefore, the controller behaves as explained in Section 5.3.1
- **Keep left:** in this mode, the left wheels are not allowed to enter the gutter, whereas the right wheels can enter but keeping at least one wheel on the floor. This mode is useful to choose the leftmost option whenever a fork is detected (Figure 5.8) .
- **Keep right:** this mode is symmetric with respect to the keep left mode. Therefore, this mode is useful to choose the rightmost option whenever a fork is detected (Figure 5.9).
- **Recovery mode:** in this mode, the robot would try to escape from the gutter once it has fallen inside. In this mode, the robot is allowed to enter the gutter with freedom. However, due to the cost involved in it, the natural behavior of the robot will be to reach a normal state (with all wheels on the floor) as soon as possible.

The whole navigation stack has been tested in the Mercat del Born scenario during September 2017, allowing to navigate in this mode the full scenario.

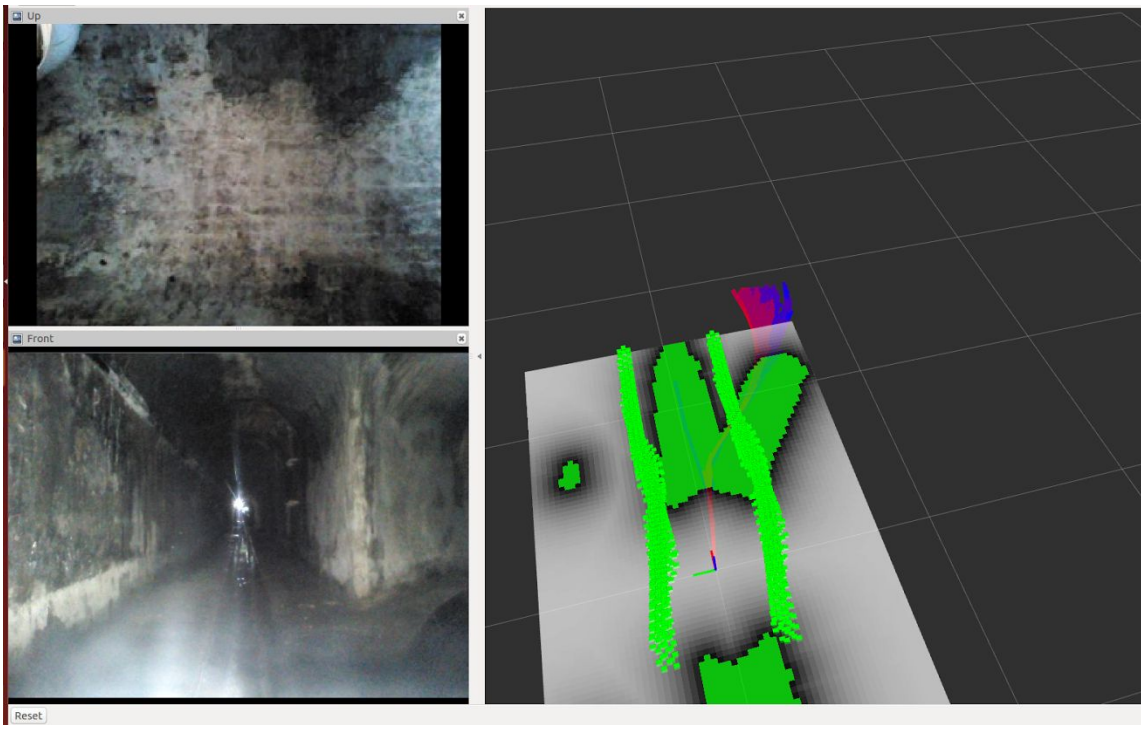

Figure 5.8. Trajectory to keep the same direction in a fork with the proposed local planning method. In this case, the keep left mode was activated

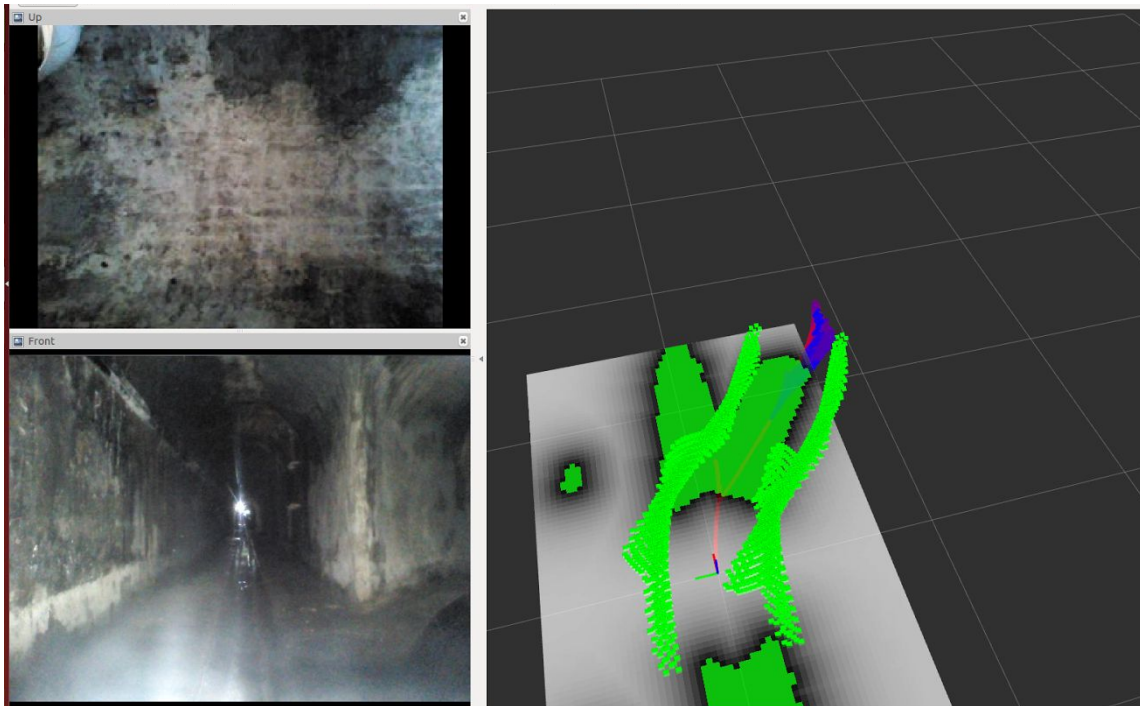

Figure 5.9. Planned trajectory to change direction in a fork with the proposed local planning method. In this mode, the keep right mode was activated.

6. Perception

According to the Challenge document [ECHORD++, 2014], a set of perception functionalities are required related to the monitoring of sewers:

- Providing 3D scanning data
- Sewer map building
- Sewer elements localization
- Sewer serviceability inspection
- Structural defects inspection

In this document we describe the functionalities as they are for this second iteration of the robotic solution. Some of the components were already described in Deliverables D28.1 and D28.5. We describe all of them here for completeness.

6.1. Providing 3D scanning data

The developed sensor payload using RGBD sensors (see Fig. 6.1) is able to provide 3D scanning data. At full resolution, the 7 cameras are able to provide point clouds of nearly 1.000.000 points per scan at 30 Hz (see Fig. 6.2). These cameras, because they have no moving parts, are able to produce these scans without requiring the robot to stop. And while the robot carries illumination devices, it could even provide these data without illumination.

Figure 6.1: The robot carries 7 RGBD cameras.
One on the top, and 3 cameras on the front and on the rear of the robot

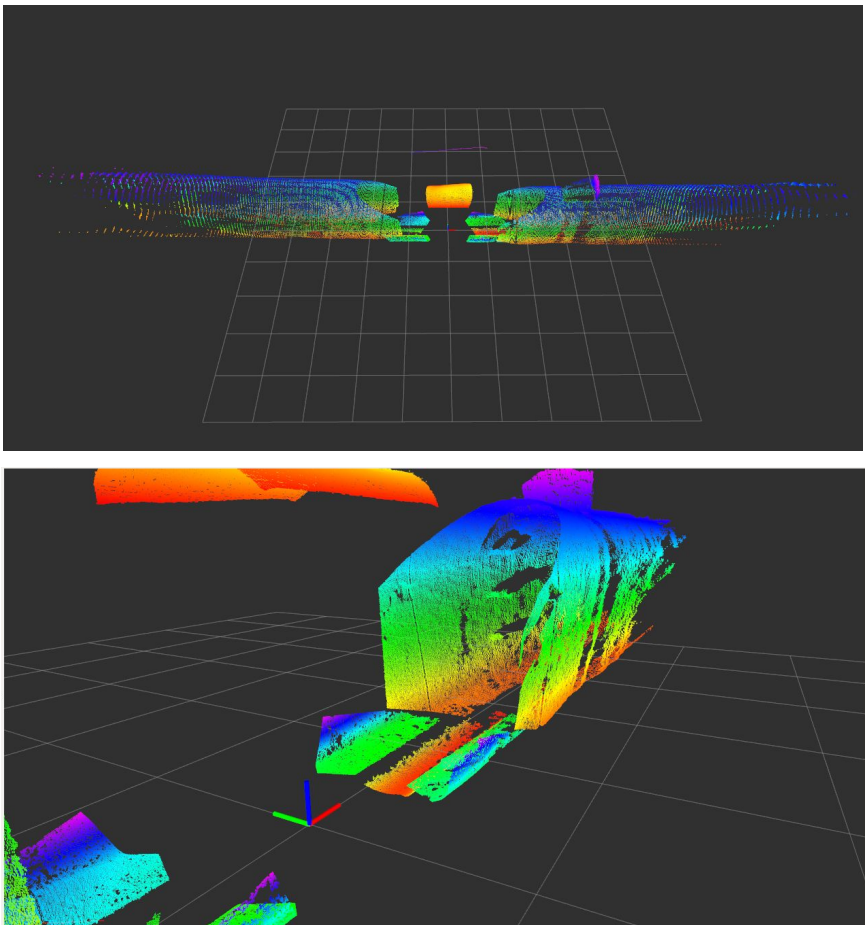

Figure 6.2: The point clouds directly generated by the sensors. Top: general view (the robot is in the center, each square in the grid represents a 1 m side square). Bottom: close-up view.

6.2. Sewer Map Building

The sensorial system described above 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 6.3 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.

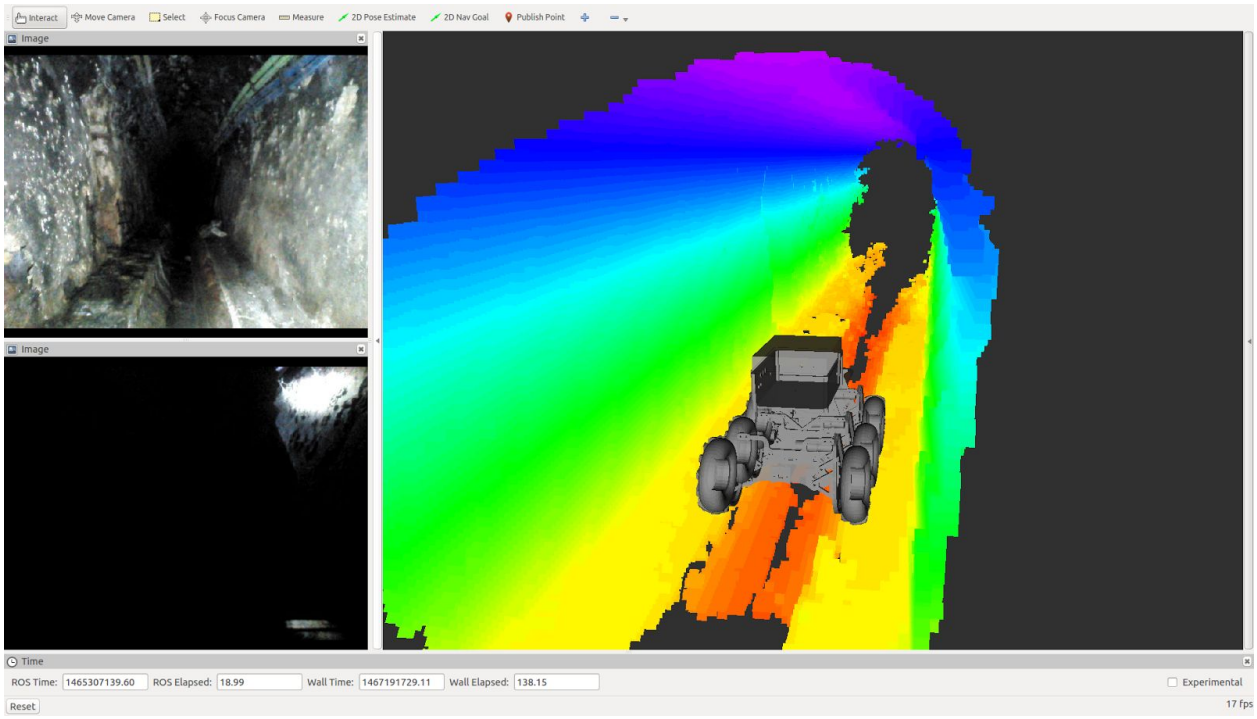

Figure 6.3. Different views of the operator while command the SIAR platform.
Left: first-person perspective (front and rear cameras). Right: third-person perspective.

Finally, these maps, combined with the localization outputs (Section 5.2), can be georeferenced into the sewer network.

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

6.2.1. 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 will be 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.

Figure 6.4 shows the different stages of the local 3D mapping process. As previously introduced, the main inputs of the system are a single point-cloud provided by the RGB-D sensors and the estimated translation since last map update given by the visual odometry system detailed in Section 5.1. With this information the point-cloud is aligned with the current map so that it can be merged with it.

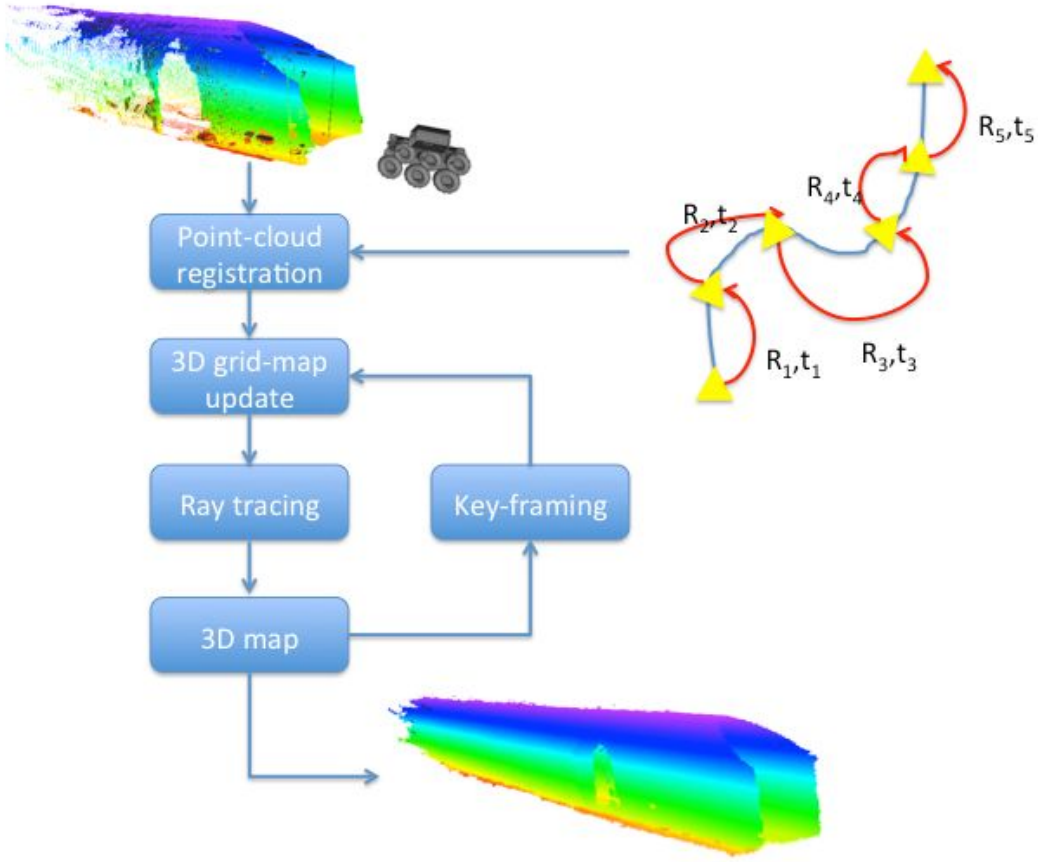

Figure 6.4. Overview of the local 3D mapping technique developed in SIAR.

The local map is represented as a 3D occupancy grid that stores the probability of 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.

Finally, the motion of the robot is checked in order to detect a new key-frame position. If the robot moves more than a given threshold, a new key-frame is added and all the map points are translated to the reference of the new key-frame. This method allows reducing the computational requirements for map building and map representation.

This system was already deployed in Phase I and described in D28.1. Figure 6.5 shows the local 3D reconstruction of Valencia Street. The map shown is only 50 meters of the total 130 traversed with robot during mobility test in Barcelona. It can be seen how the structure is also very clear locally and how the map diverges slowly.

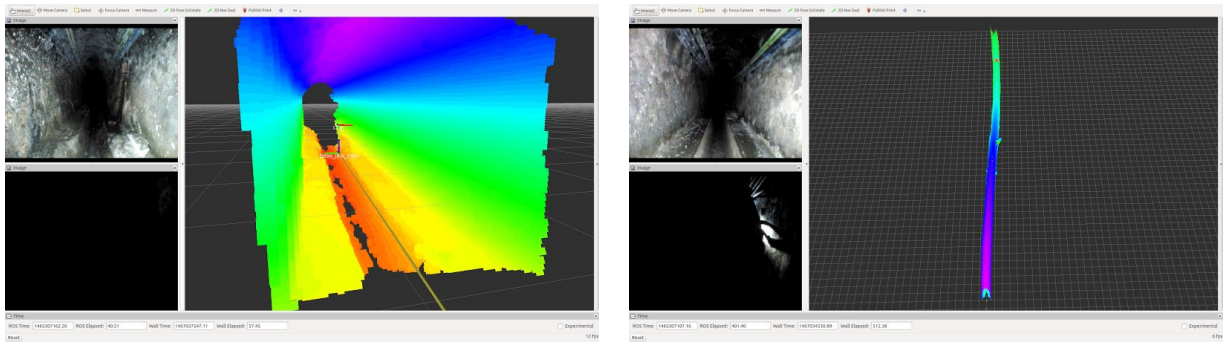

Figure 6.5: 3D map reconstruction of a 50 meters long section of Valencia sewer.

6.2.2. 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 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 (presented in next section) 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 5.1). 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 optimize 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.

Figure 6.6 shows the original graph computed with the visual odometry. The figure also shows the estimated trajectory once the optimization has been performed over the whole trajectory. In this experiment the robot traversed approximately 400 meters, then returning to the starting point. A couple of manholes located at the beginning of the trajectory were used to close-loop the pose-SLAM and to rectify the trajectory.

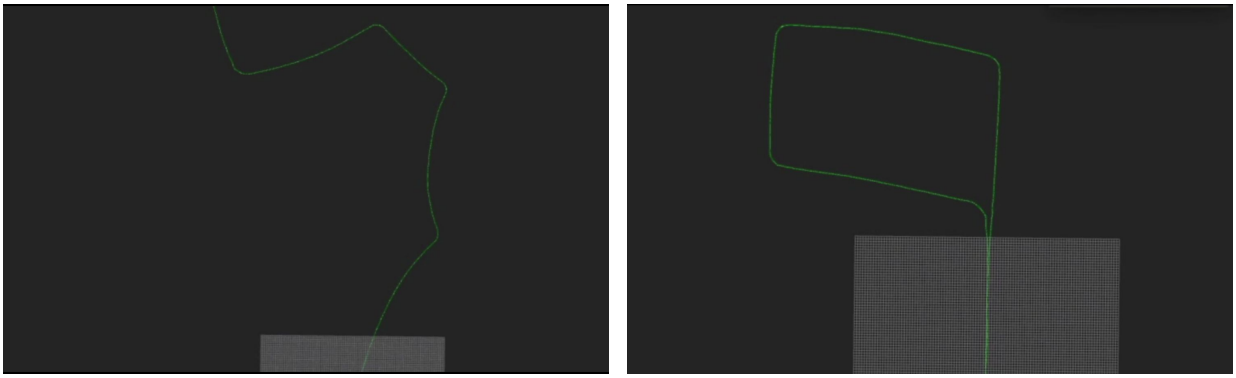

Figure 6.6: Optimization process over a real trajectory (400 meter long approximately). Left: original trajectory computed based on visual odometry. Right: obtained trajectory after non-linear optimization using manholes to detect close-loops

The outcome of the previous optimization will be a globally consistent trajectory. Now we can perform automatic loop-closing detection on the RGB-D data, based on different approaches as visual place recognition or scan matching. This approach implements a massive scan-matching process among all the poses that fall within a given search radius.

This loop-closing detection will add new constraints among the robot poses. However, this approach goes one step forward and also takes into account a different type of constraint as the usual transform between poses, so that we can optimize the alignment between the point-clouds directly into the optimizer.

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 6.7 shows a 3D reconstruction of the of sewer during the trajectory presented in Figure 6.6. It can be seen in Figure 6.7 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 6.7: A section of the global 3D reconstruction of the sewer

6.3. Sewer Elements Location

An additional required functionality is the localization of sewer elements, like manholes, inlets, crossings and others. Detecting those elements can be based on a geometric analysis of the 3D local maps described above, considering the typical geometry of the elements. However, there can be subtle differences between instances of the same element. Also, it is not straightforward to define precisely all the elements that can be found, and this is difficult to extend. Thus, we have developed an initial system based on training, in which exemplar (depth) images of the elements are used to teach the system to detect elements of interest. The system was described in D28.5 and it is summarized here.

As a first step, we use Convolutional Neural Networks (CNNs) for the detection of manhole elements from depth images. CNNs seem a good option as they have been extensively used for image classification purposes in last years with great success.

In this first iteration we have focused on manholes because of their particular shape, as they break with the uniformity of the gallery ceiling which might simplify its detection (see Figure 6.8). However, we consider that a similar method could easily be used for detecting other types of elements such as forks and inlets without any trouble. The reader is referred to [Alejo et al., 2017] for more details on the detection system.

Figure 6.8. View of the sewer gallery from the depth camera pointing towards the ceiling.
Left: regular gallery ceiling. Right: manhole.

The CNN architecture designed for manhole detection is shown in Figure 6.9. The input images are downsampled to 80x60, as the manholes should remain detectable. This CNN has roughly 9000 parameters and its computation time is below 1.5 ms in a regular i7 laptop.

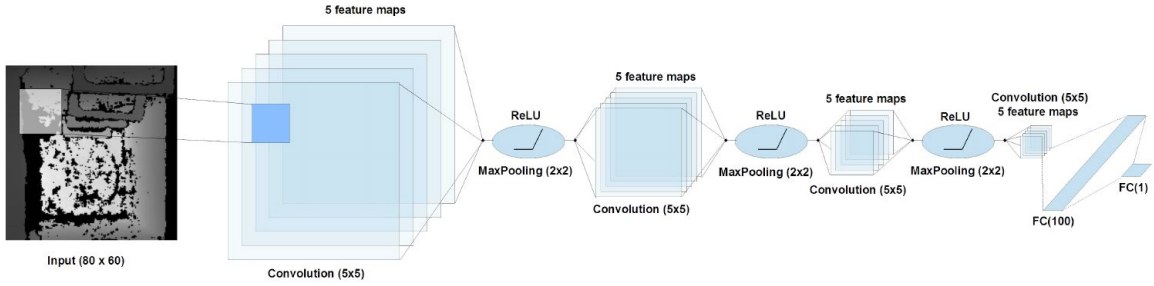

Fig 6.9. Convolutional Neural Network for automatic manhole detection based on depth images

The CNN has been trained and validated with a dataset that is composed by 40000 depth image samples with resolution 80x60 gathered from two experiments at the “Mercat del Born” on the 17th of January, 2017. Each sample integrates a label that indicates if the image contains a manhole or not. From these samples, 21000 are used to to validate the CNN and are not included into training process.

After the training process, we performed a validation experiment. The obtained detection accuracy is roughly a 96%. These results are very satisfactory, taking into account that different types of galleries visited in both datasets and that there were opened and closed manholes. Table 6.1 shows the validation results in more detail. It can be seen how the false positive rate is approximately of a 2%. On the other hand, the false negative rate is bigger, going up to a 10%. This indicates the probability of missing one manhole by considering one image. However, as the robot traverses a manhole, it can take tens of images of the manhole. Therefore, the probability of missing a manhole is much lower.

		Predicted	
		Positive	Negative
Actual	Positive	0.90	0.10
	Negative	0.03	0.97

Table 6.1: Manhole detector confusion matrix

6.4. Sewer serviceability inspection

The system should be also able to determine if the serviceability of the sewer has been reduced, raising alarms in case this occurs. This requires, according to [ECHORD++, 2014] to distinguish waste on the sewers, of different sizes.

From the local 3D maps described above, the required precision can be achieved. Figures 6.10 and 6.11 shows the 3D data and how a 5 cm tall rock can be identified visually in the 3D cloud.

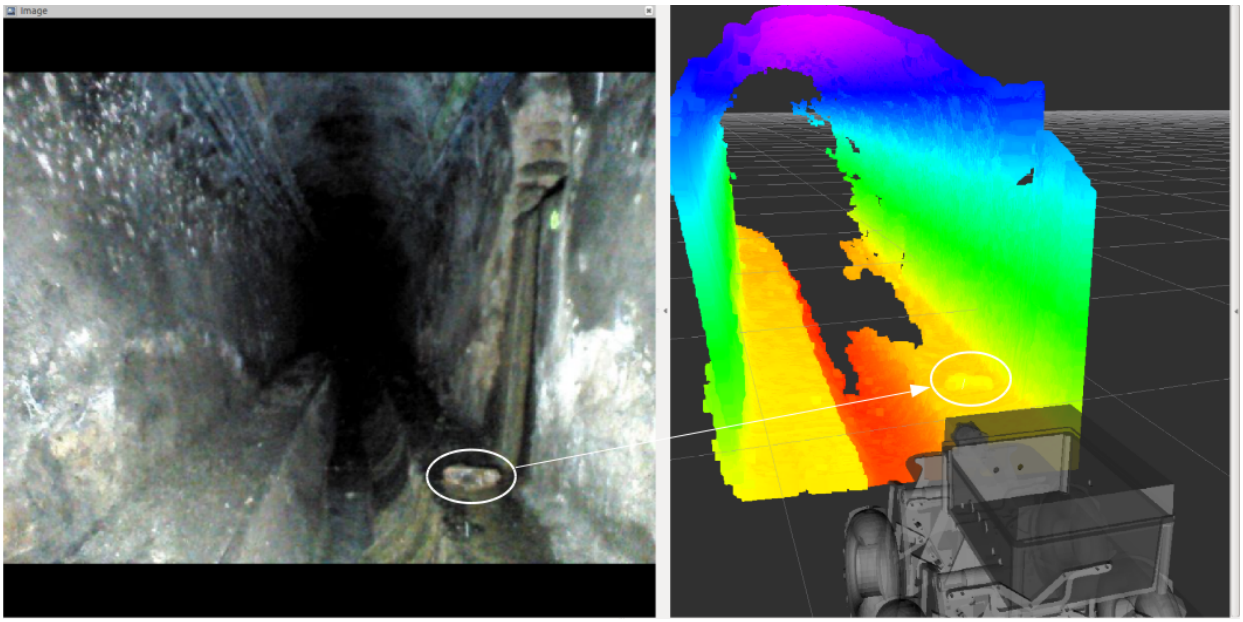

Figure 6.10: Left: a rock can be seen in the curb of the sewer. Right: 3D reconstruction. The rock is around 5 cm tall.

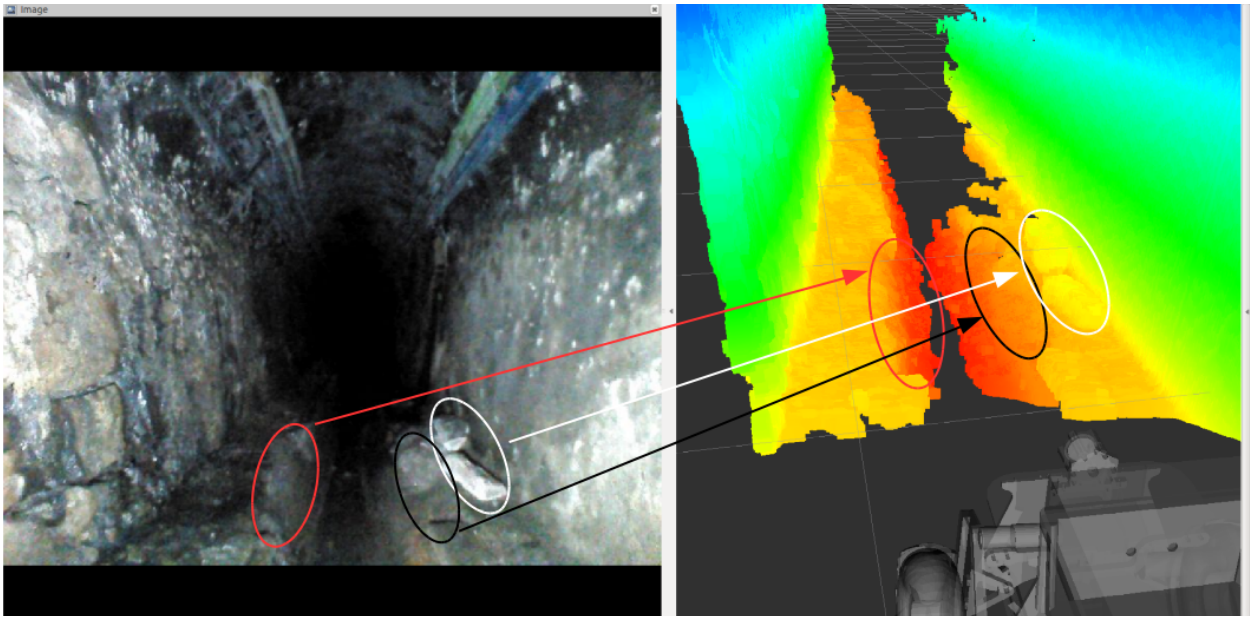

Figure 6.11: Top: the reconstructed 3D cloud. Some rocks (white) and break with losses on the curbs (red and black) can be noticed in the image in the top left, and in the cloud on the right.

6.5. Structural defects inspection

According to [ECHORD++, 2014], different critical defects have to be detected, from cracks and fractures to collapses.

The module for structural defects inspection should automatically analyze the 3D information to search for such defects and then alert the operator, who can then confirm the defect. From the local sensors on the robot, it is possible to determine if the defect is located in the vault, the walls or the floor. The information from the localization module will be used to provide the distance to the closest manhole of the defect and the position in the general GIS.

For the automatic detection of defects, we are considering the known geometry of sewers. The different sections are included in the GIS, and have a known metric shape. From this known shape, it is possible to generate artificial point clouds, as shown in Fig. 6.12.

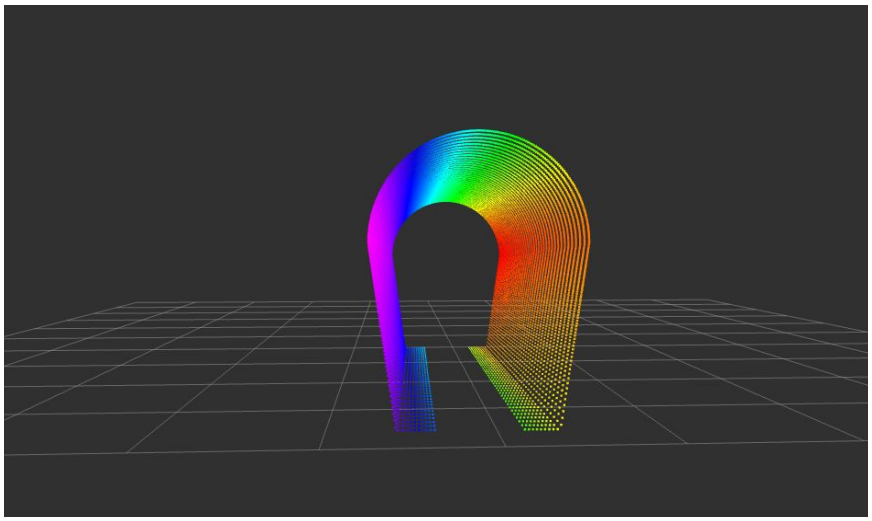

Figure 6.12: 3D point cloud of the ideal section type NT120A

The current procedure aligns, by using Iterative Closest Point matching, the virtual model with the current point cloud gathered by the cameras onboard the robot (see Figure 6.13). The section is selected by the position of the robot, but the procedure itself could be used to detect the type section in which the robot is located and actually help in the localization process.

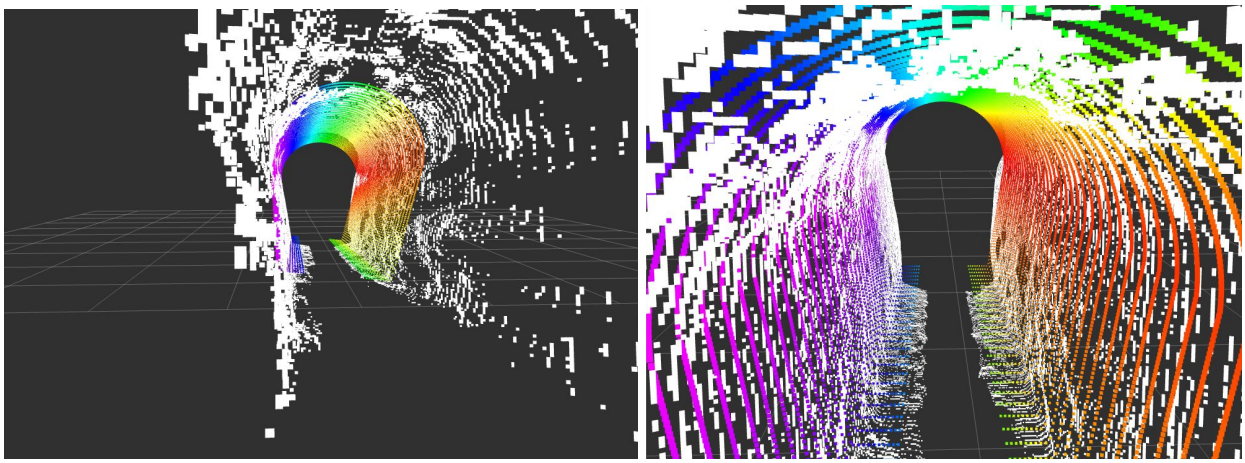

Figure 6.13: The ideal model is aligned with the current point cloud. Left and right show two different views of the alignment. This can be used to seed and analysis of potential defects.

Once aligned, the differences between the model and the real point cloud can be used to seed a search for cracks and defects. Figure 6.14 shows an example using real data from the sewers.

Figure 6.14: Overlaid on the original images some candidate zones of defects are shown. Left: some cracks on the floor and left wall can be seen. Right: a deformation on the left wall is highlighted, as well as a hole on the right (in this case, an inlet).

The current system is being refined to reduce the number of potential false positives by temporal data fusion, and using more dense point clouds from the local maps generated. The actual sections show deviations with respect to the ideal ones, and thus additional procedures are required to obtain more accurate detections.

7. Future Work

In order to achieve the final SIAR robot configuration some technical features will be added to the robot in Phase III of the project. Environmental sensors, sampling actuators, mechanical protections and software improvements are some of the features to be added/improved.

7.1. Environmental Sensors

The environmental sensors will be used to monitor the environment temperature, humidity and gases, determine the variations and alert the operator in case of danger. The included sensors will be able to monitor air and water variables of interest.

7.1.1. Air Sensors

An Air sensors setup system will be used to perform the air test. There is already a commercial system that allows the measurement of required air parameters. The Libelium Gas PRO board (Figure 7.1) together with the Waspmote board fulfil the requirements for the temperature, humidity and calibrated gases measurement.

Figure 7.1. Libelium air sensor system.

The Libelium Gas PRO system will be connected to the Waspmote developer board. The Waspmote developer board allows the users to process all the information from the sensors and transmit the processed information using different types of wired and wireless communications.

- **Temperature (ºC) sensor MCP9700A**
    - MCP9700A Specifications
        - Measurement range: -40ºC to 125ºC
        - Response time: 1,65s
- **Relative Humidity (%RH) sensor 808H5V5**
    - 808H5V5 Specifications

        - Measurement range: 0 to 100%RH
        - Operating Temperature: -40ºC to +85ºC
        - Response time: <15s
- **Carbon Monoxide (CO) sensor TGS2442**
    - TGS2442 Specifications
        - Gases: CO
        - Measurement range: 30 to 1000ppm
        - Operating temperature: -5 to +50ºC
        - Response time: 1s
- **Hydrogen sulphide (H_2S) sensor TGS2602**
    - TGS2602 Specifications
        - Gases: $C_6H_5CH_3$, H_2S, CH_3CH_2OH, NH_3, H_2
        - Measurement range: 1 to 30 ppm
        - Operating Temperature: +10 to +50ºC
        - Response time: 30s
- **Methane (CH_4) sensor TGS2611**
    - TGS2611 Specifications
        - Gases: CH_4, H_2
        - Measurement Range: 500 to 10000ppm
        - Operating Temperature: -10 to 40ºC
        - Response time: 30s
- **Oxygen (O_2) sensor SK-25**
    - SK-25 Specifications
        - Gases: O2
        - Measurement range: 0% - 30%
        - Operating temperature: +5 to 40ºC

        - Response time: 15 s

- **Lower explosive limit (LEL)**
- **Volatile organic carbons (VOCs) sensor MiCS-5524**
    - MiCS-5521 Specifications
        - Gases: CO, Hydrocarbons, Volatile organics Compounds
        - Measurement range: 30 to 400ppm
        - Operating temperature: -30 to +85ºC
        - Response time: 30s

7.1.2. Water sensors

A Water sensors setup system will be used to perform the required water test. There is already a commercial system that allows the measurement of required water parameters. The Libelium Smart Water sensor board (Figure 7.2) together with the Waspmote board fulfil the requirements for the project with the inclusion of the following probes:

- **Temperature (ºC) sensor PT1000**
    - PT1000 Specification
        - Measurement range: 0 to 100ºC
- **PH sensor PH**
    - Libelium PH sensor Specifications
        - Measurement range: PH 0-14
        - Temperature range: +0 to 80ºC
        - Response time: < 60s
- **Conductivity sensor**
    - Measurement range: 0 to 60,000 uS/cm
- **Turbidity Sensor**
    - Libelium Turbidity Sensor Specifications
        - Range: 0 to 4000 NTU
        - Temperature range: -10 to +60ºC

The Libelium Smart Water sensor system will be connected to the Waspmote developer board. The Waspmote developer board allows the users to process all the information from the sensors and transmit the processed information using different types of wired communication.

Figure 7.2. Libelium Smart water system

The system will use a peristaltic pump to take water from the sewer and inject the sewer water into the water sensors assembly. Figure 7.3 shows the D1/PHS SUPERPRO water sample system that allows to control the PH and Temperature. The idea is to create a more complex sampling sensor system using the same principle.

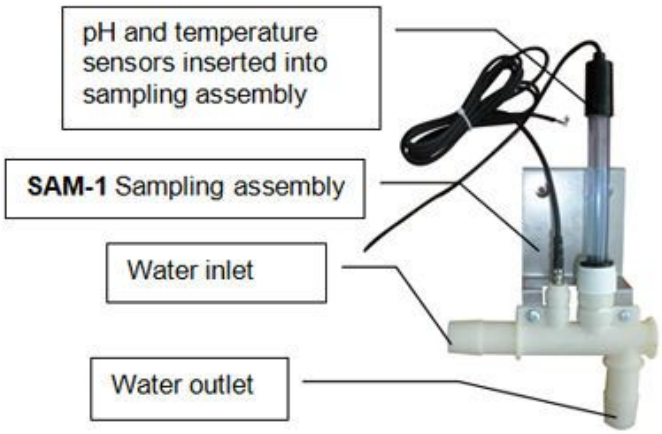

Figure 7.3. Water sensor system.

7.2. Sampling Actuators

The SIAR robot will be able to collect samples from the environment. The robot will carry three different sampling systems:

- Water Sampling. It will transport a system able to collect water from the sewer and store it in 300 ml reservoirs;

- The system will use a Peristaltic pump to collect water from the sewer and using up to 8 water solenoid electric valves redirect the water to small 300 ml reservoirs. Each reservoir will be equipped with a level sensor to determine if the reservoir is full.
- Air Sampling. It will carry commercial air capsules that will be used to collect samples from the air;
- Sediments sampling. It will carry a device able to collect samples from the sediments.

7.3. Mechanical Protection

Mechanical protection must be used wherever mechanical components are at risk of failure due to contact with environmental elements. For the new prototype a protection must be created to protect the linear actuator and its potentiometer from shocks with the gutter structure or with debris. The linear actuator and its potentiometer are located at the bottom of the frame (See Figure 7.4).

Figure 7.4 - Linear motor and its potentiometer and the bottom of the robot.

7.4. Automatic defect detection

The current solution based on sewer model matching has demonstrated to be useful and accurate enough. Nevertheless, comparing high resolution point clouds is computationally demanding, so that this utility consume a considerable amount of computational resources of the onboard computer.

Future work will consider extending the defect detection algorithm with new methods to optimize the matching search. Thus, approximate nearest neighbours (ANN) techniques can be apply to significantly reduce the total time required for point cloud alignment. Also, different heuristics can be introduced to bias the search so that we can reduce the matching time.

7.5. Automatic sewer type detector

Although the sewer defect detection system is able to automatically switch between different sewer sections, it would be desired to have a system able to automatically recognize the type of sewer gallery or if the robot is in a fork or in a curv. Future work will consider training a classifier based on deep learning able to infer the type of section and the general situation: straight line, fork or curve. This information can be also very beneficial for localization.

7.6. Installation of GPU onboard the robot

Given the extensive use that SIAR project is making of Deep Learning Approaches, we are considering installing a small-size GPU onboard the robot, as a NVIDIA Jetson X1 or X2. These boards are relatively low power consuming and small, so the impact in the robot would be small. However, they can boost the speed of deep learning neural networks.

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.

[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.