



Sewer **i**nspection **a**utonomous **r**obot

D28.5 – Prototype for Sewer Inspection

SIAR Consortium

IDMind (IDM), PT

Universidad de Sevilla (USE), ES

Universidad Pablo de Olavide (UPO), ES

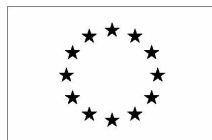


Table of Contents

1. Introduction	2
2. Robot Platform Kinematics	3
2.1. Frame	3
2.2. Width Adjustment Mechanism	4
2.3. Traction System and Suspension	5
2.4. Payload Electronics	6
2.4.1. Cameras' Disposition	8
3. The robotic solution for deploying/collect the repeaters	10
4. Navigation and Inspection Capabilities	13
4.1 Navigation	13
4.1.1 Local Navigation	13
4.1.2. Planning maneuvers in the presence of forks	15
4.1.3. Improved kinematics	17
4.2 Inspection Capabilities	21
4.2.1 Local 3D reconstruction	21
4.2.2. Global 3D reconstruction	22
4.3 Robot localization in the sewer system	27
4.3.1. Graph-based Localization	28
4.3.2 Field experiments	30
References	34



1. Introduction

The Phase I prototype was used to test the system in the sewers of Barcelona. The tests allowed the consortium to take conclusions about the best options to create a Sewer Inspection Autonomous Robot.

The Phase II robot results in a robot with the same proven type of locomotion using additional features. These features allow an easier robot deployment and will increase the navigation and maneuverability capabilities of the robot while changing direction inside the galleries.

With the results obtained from experiments with the prototype, a new and better camera positioning was defined and implemented.

The wireless repeaters and the robotic arm had to be changed to be able to be carried by the new robot.

This report describes the adopted ideas for the new SIAR robot, the new platform, the arm, the repeaters and the navigation and the inspection capabilities.

2. Robot Platform Kinematics

A new iteration of the previous model of the robot was developed. In this model, improvements were made. A width adjustment mechanism was developed in order to increase the adaptability of the robot to the sewer architecture. Both the frame and the traction system suffered changes in order to accommodate the width adjustment mechanism. The width adjustment is actuated with the use of a linear motor. A metallic frame is used and an arrangement made of gears and sprockets is used to propel the robot. Figure 2.1 shows a CAD of the fully assembled SIAR robot.

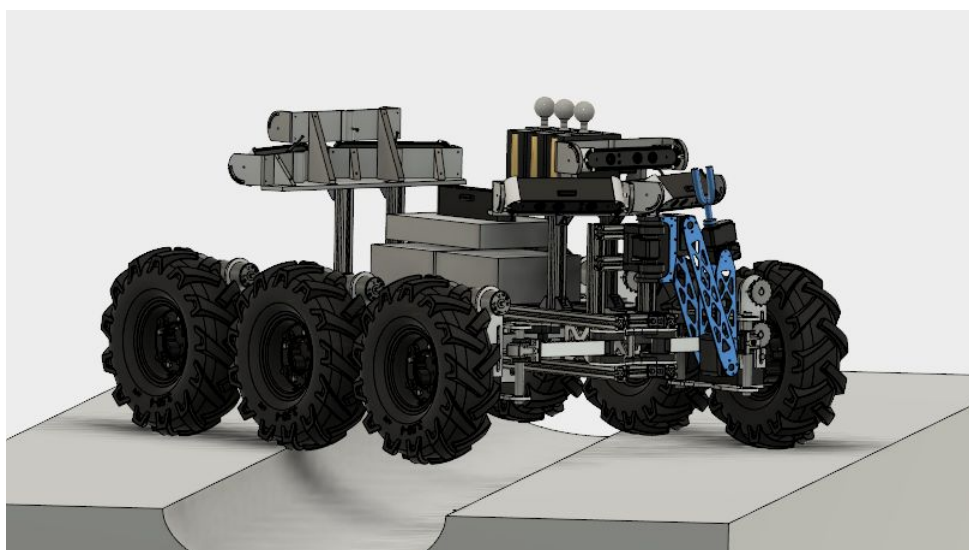


Figure 2.1. Fully assembled SIAR robot CAD.

2.1. Frame

The new frame used in this prototype is mainly made of aluminium and POM. The frame intends to accommodate the payload and the width adjustment mechanism. The payload is connected to the upper part of the frame. The linear motor, that actuate the width adjustment mechanism, is connected to the bottom of the frame. 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. In Figure 2.2, the upper (left) and lower (right) views of the frame are presented.



Figure 2.2. SIAR robot frame upper view (left) and lower (right).

2.2. Width Adjustment Mechanism

In order to improve the robot adaptability to different structural changes in the working environment, a width adjustment mechanism was developed. This mechanism allows to vary the width of the robot by mean 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. To transmit the movement from the motor/carriage to the wheels, metallic rods are used. Rod ends are used to allow the rotation of the rods at its ends. Each one of the six wheels of the robot are connected to the frame by mean of two arms. At each end of the arms there are bearings to allows 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 grants 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. In Figure 2.3, two width configurations are presented. With this mechanism the width of the robot can vary from 500mm (minimum) to 700mm (maximum).

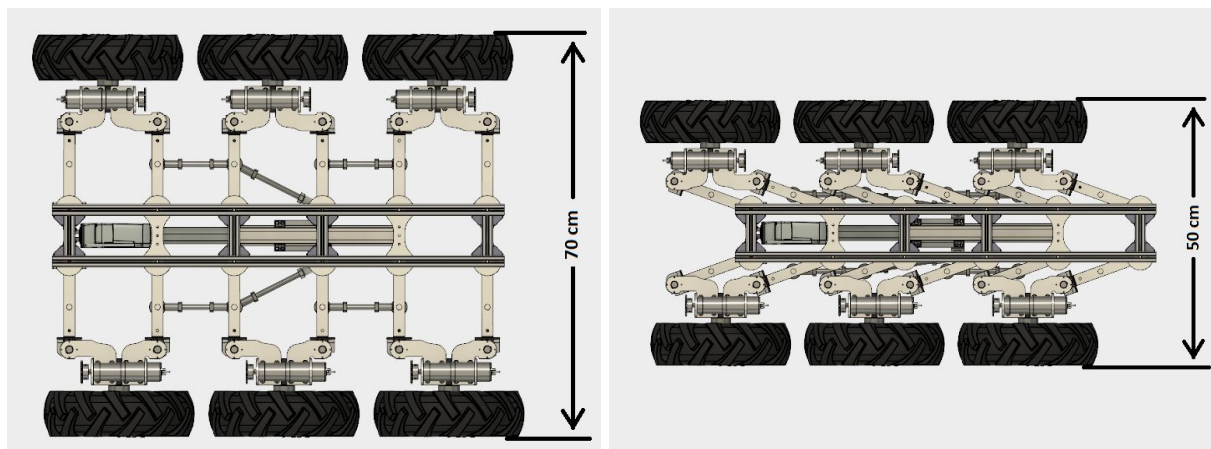


Figure 2.3. SIAR adjustment mechanism. Left: maximum width. Right: minimum width.

2.3. Traction System and Suspension

The new prototype still uses a configuration composed by six tractive wheels. This systems showed to be effective in previous experiment in the sewer. The main differences in the new prototype are the way the output torque of the motors is transmitted to the wheels and the presence of a suspension connected to the wheels.

The new tractive system is composed by two sprockets and a worm gear set. The two sprockets are connected by a chain. One of the sprockets is connected to the motor shaft, the other one is connected to another shaft that contains the worm gear. The worm gear transmits the motion to a gear that is connected to the wheel. This way, the torque is transmitted from the motor to the wheel. 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 2.4.



Figure 2.4. Implemented traction system.

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 mechanism is presented in Figure 2.5, with two different suspension travel values.

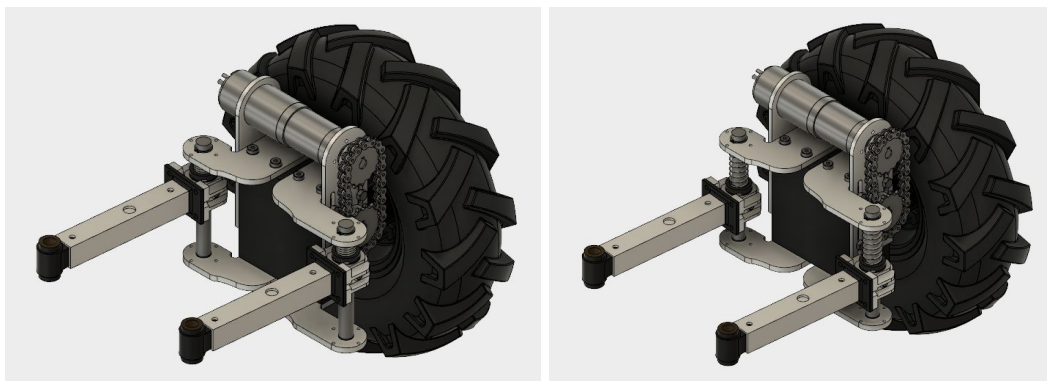


Figure 2.5. SIAR suspension.

2.4. Payload Electronics

The system is composed by the following components:

- **NUC i7.** This is the main processor that will get the information of the sensors inside the SIAR. It will communicate with the base station through a 2.4GHz and will send the commands to the SIAR main control board and gather the data from the encoders.
- **Four Astra S cameras and three Astra cameras.** The cameras will be the main sensor for navigation, localization and structure analysis.
- **SIAR Motor control board.** Is connected to the six motors of the platform and offers the encoder data to the NUC i7. It is connected to the NUC i7 through the USB hub. The motor controller is the same used in the Phase I prototype.
- **SIAR Linear Motor board.** Is an add-in board that is connected to the SIAR Motor control board and allows the control of the width of the robot.
- **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.

- **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 will allow to control the platform taking into account dynamic and kinematic effects. It also will improve the localization of the platform. It is connected the through the USB-HUB.
- **Microhard Nvip 2400 WiFi link.** This link offers a high bit rate (up to 45Mbps) but lower range (up to 10km 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 odometry information. It is connected to the NUC through an Ethernet connection (RJ-45).

Figure 2.6 shows the adopted electronic architecture for the II Phase electronics.

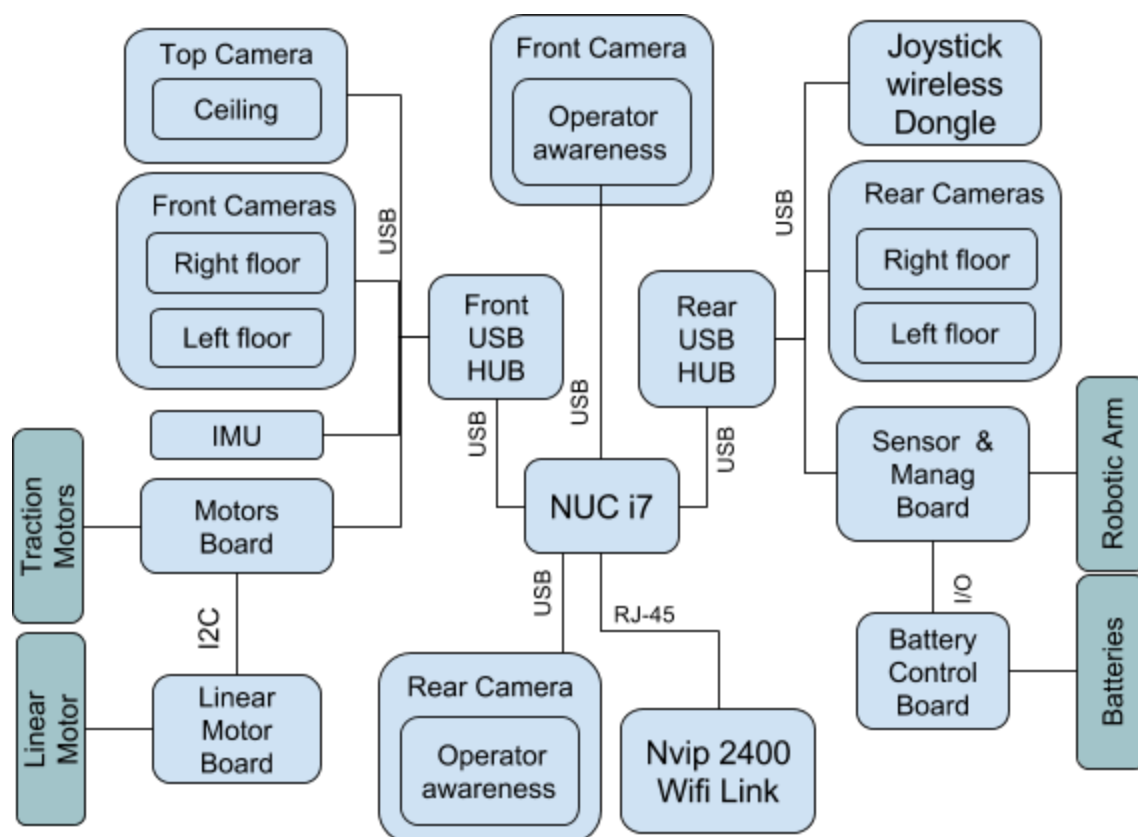


Figure 2.6. Electronics' architecture.

The individual boards architecture and a more exhaustive description will be described on the robot report D28.6.

2.4.1. Cameras' Disposition

Experimental results have demonstrated the need of sensing the closest areas of the robot in both front and rear sides in order to properly detect possible obstacles and plan the required actions to avoid them. Figure 2.7 shows the onboard cameras disposition as installed in the Phase I prototype. This configuration was successfully tested during the field experiments at the “Mercat del Born” that were carried out on the May 22-23, 2017.

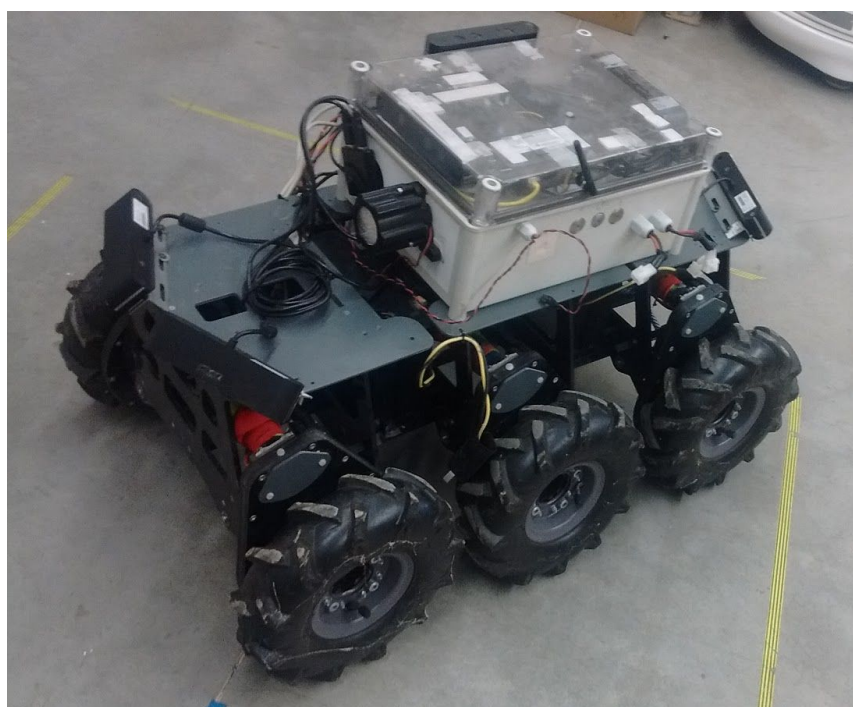


Figure 2.7. Cameras' disposition in the prototype of Phase I

The following cameras are used:

1. Four Astra S cameras (two front and two rear) tilted 45 degrees (Figure 2.8 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 2.8 right) is possible to have a combined 110° field of view in each direction (backwards and forwards). Result 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.

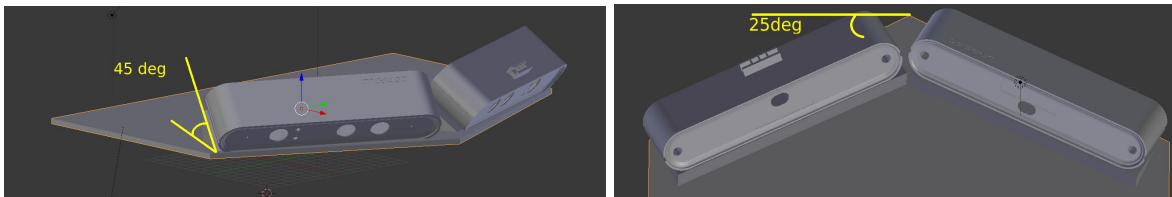


Figure 2.8. Left: tilt angles of the sensors.

Right: top view of the sensing system, the yaw with respect to the horizontal of the robot is marked.

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 8m) that are used by 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.

Figure 2.9 depicts the camera's FoV cones.



Figure 2.9. Proposed disposition of the cameras that compose the sensing system.

3. The robotic solution for deploying/collect the repeaters

The main goal of the robotic arm is the deployment and collection of the wireless repeaters in the sewer. The wireless repeaters intend to increase the signal range.

The wireless repeaters (from D28.3) suffered some changes in order to facilitate their manipulation with the robotic arm. The nVIP2400 wifi devices were disassembled and new batteries were selected in order to reduce the overall size and weight. The electrical components and the battery were then assembled together in a plastic box, as shown in Figure 3.1. The box has a handle on the top to allow the coupling with the claw of the robotic arm. Table 1 shows the items that compose the repeater, their consumption or battery capacity and their weight.



Figure 3.1. Components installed inside the wireless repeater.

Item	Power (W)/ Capability(Wh)	Weight(g)
LiPo Battery (3S)	1450mAh@11.1V (16.1Wh)	97.3
<u>nVIP2400</u> (without box)	6W (max)	100.1
Enclosure + Handle		90
Antenna		30
Total	>2.5 hours of operation	317.4

Table 1. Wireless repeater components

With the new battery the repeater power autonomy will be close to 2.5 hours. Please note that this consumption is calculated in the worst case scenario with the repeater working at its maximum and letting the batteries to be discharged at 80% of their capacity, following the manufacturer's guidelines. To increase this autonomy, the repeater will be equipped with a small magnetic reed sensor that will connect/disconnect the battery power from the wireless nVIP2400 WiFi device. The top of the wireless repeaters box installed on the robot will contain magnets that will deactivate the power from the repeaters. When the repeater is taken from the robot, by the robotic arm, the reed sensors stops sensing the magnet and the battery is connected to the wireless wifi device inside the box. In this way, it is expected that each repeater will have power enough for performing a full mission of operation. Figure 3.2 shows a fully assembled wireless repeater.



Figure 3.2. Developed wireless repeater.

The robotic arm has five degrees of freedom and it is actuated with the use of five Herkulex servos of different forces. 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. When completely extended the robotic arm is able to reach a distance of 495mm. Figure 3.3 shows the CAD image of the developed SIAR robotic arm.

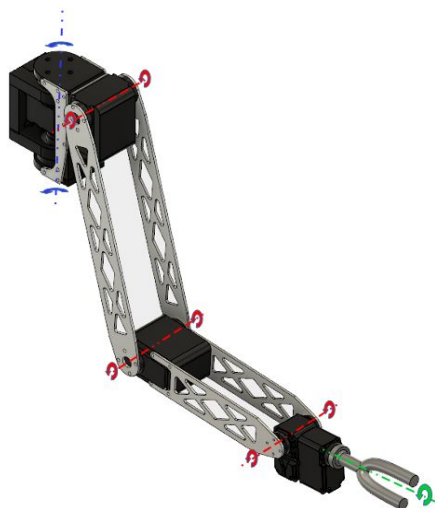


Figure 3.3. SIAR robotic arm degrees of freedom.

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 3.4) and deploy it on the ground (Figure 3.5). In the inverse is also valid, this is, the robotic arm also has the ability to collect the repeaters.

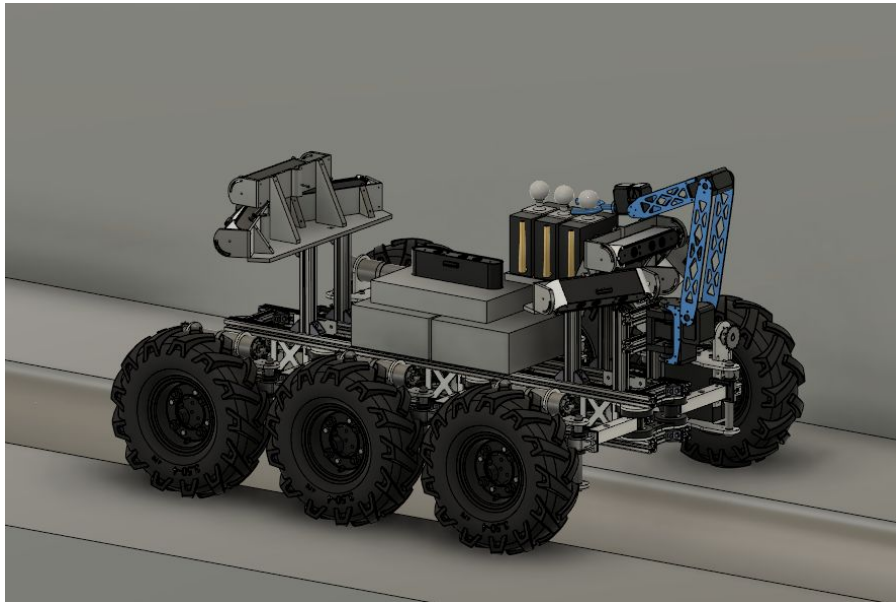


Figure 3.4. Taking the wireless repeater from the robot top.

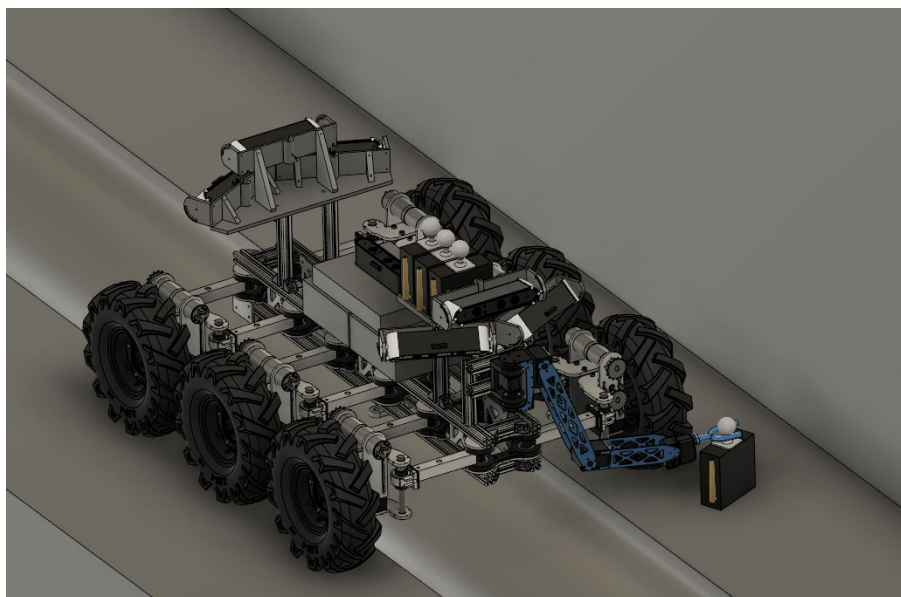


Figure 3.5. Deployment of the wireless repeater on the sewer floor.

4. Navigation and Inspection Capabilities

At the end of Phase II, the robot should be able to autonomously navigate through the sewers, sending the operator precise information about the estimated location of the robot. To this end, we have developed several software packages that are detailed here. Moreover, the new design of the robot will allow it to easily rotate over the gutter, and thus to change direction of the movement whenever a fork is reached. The advances in the navigation task are described in Section 4.1. Section 4.2 describes the advanced carried out in the inspection capabilities. Finally, Section 4.3 shows the details of the localization system. All of the work presented here has been developed with real data gathered from the sewers at the “Mercat del Born” or even tested in real-time there.

4.1 Navigation

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 4.1.1) 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 4.1.2. The improved kinematics of the Phase II robot (Section 4.1.3) offers additional degrees of freedom that can be used at both levels.

4.1.1 Local Navigation

The depth images obtained by the front and rear cameras (six in total) are combined and processed in order to obtain an altitude map of the environment close to the robot. This is done in real time at a frequency of 10 Hz. Then, this altitude 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 4.1 shows an altitude map obtained from an experiment in the sewer mockup of the University Pablo de Olavide.

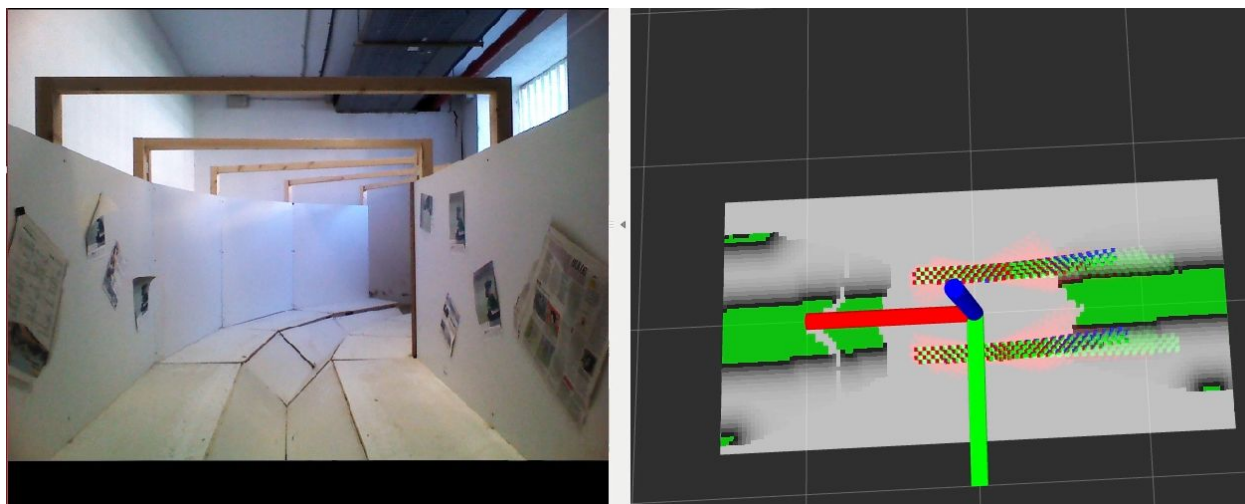


Figure 4.1. Left: Picture obtained from the operator awareness camera. Right: Associated traversability map, with untraversable zones in green (note the detected gutter in the middle) and checked trajectories: in red untraversable, in green traversable and best trajectory in blue.

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 command given by the operator and then evaluates similar commands. To this end, the module predicts the trajectory of the robot during the next T seconds and checks whether the wheels of the robot will pass through an untraversable zone or not. In that case, the checked command is discarded. Figure 4.1 (right) depicts the footprint of the wheels of the robot for the tested predicted trajectories by the module.

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

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

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

Finally, the trajectory with lowest rank is kept and its related command is sent to the controller of the robot. The best trajectory is marked in blue in Figure 4.1 (right). This operation is also repeated at a frequency of 10 Hz.

The local navigation module has been successfully tested in both a custom made mockup of the sewers that is placed at the University Pablo de Olavide and the real sewers of "Mercat del Born" in May, 2017. Figure 4.2 represents two snapshots of these experiments. In these experiments the platform was able to complete more than 50% of the track in assisted teleoperation mode.

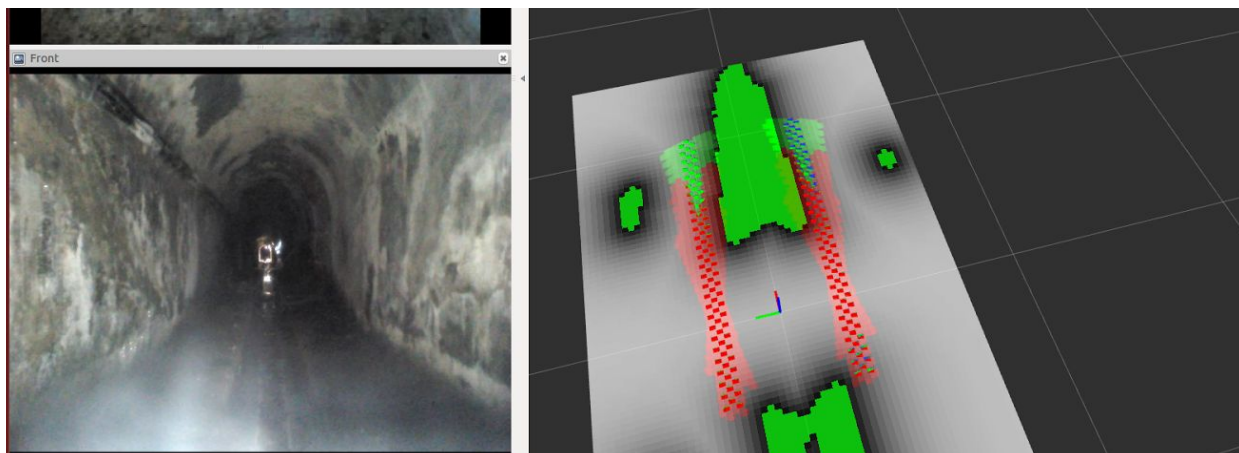


Figure 4.2. Left: Picture obtained from the operator awareness camera. Right: Associated altitude map and checked trajectories: in red untraversable, in green traversable and best trajectory in blue.

4.1.2. Planning maneuvers in the presence of forks

Even though the local navigation module has proven its reliability, 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.

Therefore, whenever the local navigation module detects the presence of a fork, another module will take the responsibility of making the maneuver to let the robot go through the fork safely in the desired direction. The robot will wait for the operator to select the desired direction with the controller in Assisted Teleoperation mode (see D28.4, Section 2.3.4 for more details) or will retrieve the desired direction from the inspection plan in Autonomous Mode. To this end, a Dynamic A* based planner has been developed and tested with the help of the data gathered during the visits to the sewers. The main characteristics of the algorithm can be summarized as follow:

- The graph is incrementally generated by integrating a kinematic model of the robot in a set of predefined maneuvers that can be configurable via input file or randomized. In particular we have found that adding pivotal rotations of the robot help to obtain safe and feasible trajectories.
- The allowed positions of the robot have been set to the ones that are dynamically stable. This has been calculated with the help of physical simulations and contrasted in real experimentation in the sewers. For example, the robot is allowed to have only the wheels of one diagonal on the floor in the riskier maneuvers.

- The planner will try to find a safer as possible path. Difficult states (i.e. states with few wheels on in contact with floor) are allowed only when necessary.
- At the moment we employ the euclidean distance as the heuristic function. However we apply weighted A* [Rivera et al., 2013] for speeding up the trajectory generation process. At the moment the module is capable of generating trajectories in few seconds.

Figures 4.3 and 4.4 represent obtained trajectories with real data for both changing direction and going straight when a fork is detected. The computation time was of about 2s in each case. However, we are also studying other methods for planning by combining bi-RRT [Kuffner et al., 2000] and t-RRT [Jaillet et al., 2008] variants, as optimality is not as important as safety for this kind of maneuvers. In this way, we expect to speed up the trajectory generation so we can replan the trajectories while performing the fork negotiation maneuvers.

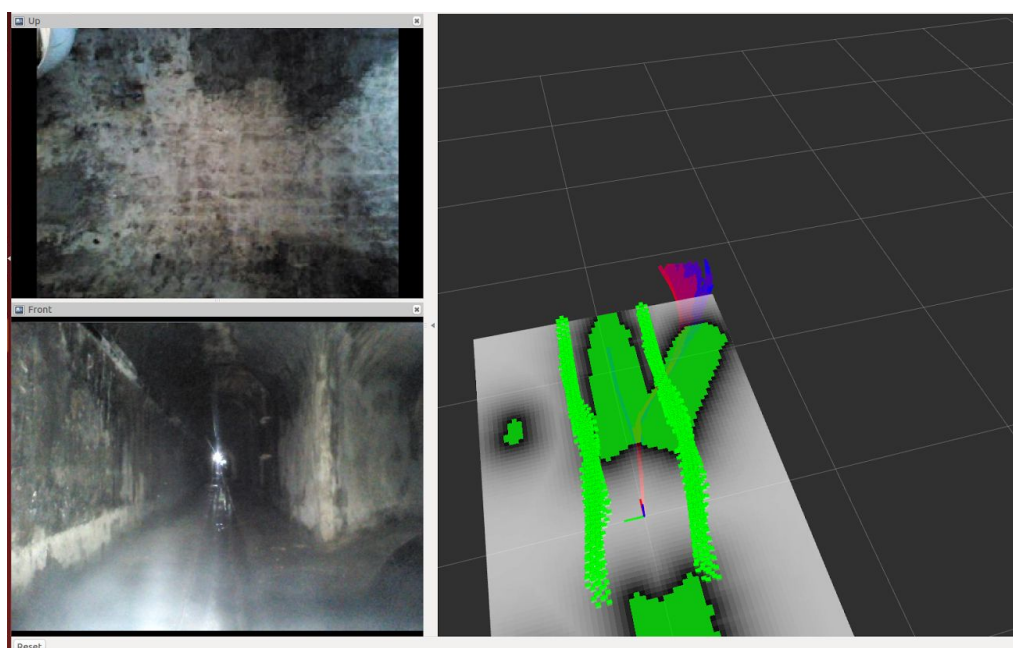


Figure 4.3. Planned trajectory to keep the same direction in a fork with the proposed planning method.

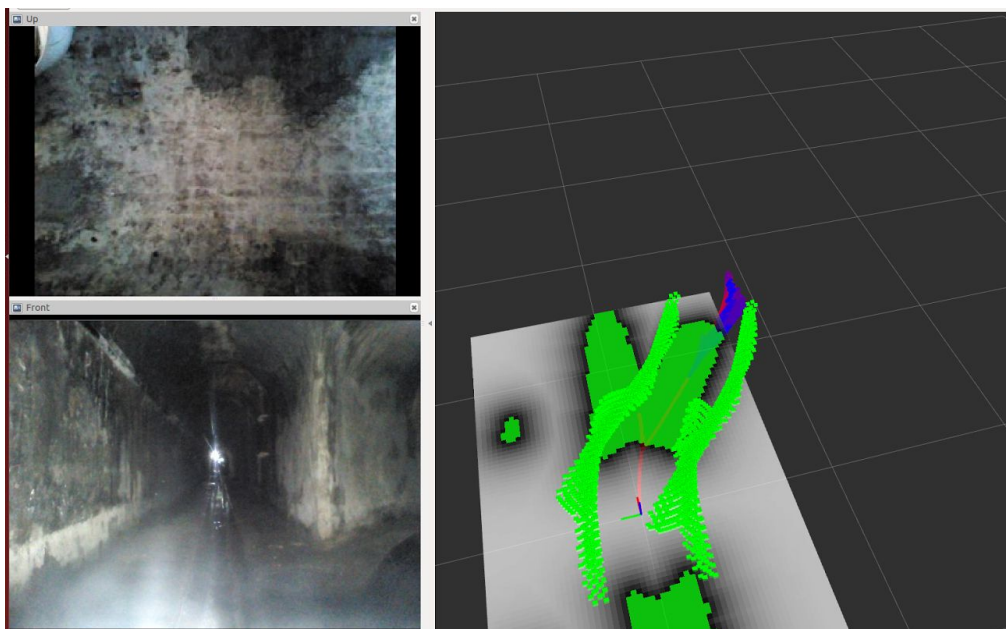


Figure 4.4. Planned trajectory to change direction in a fork with the proposed planning method.

4.1.3. Improved kinematics

In the Phase II SIAR robot is able to change its width and at the same time use this feature to change its center of mass. Figure 4.5 shows the robot passing the gutter while changing the center of mass. This feature allows the robot to safely pass the gutter without falling inside the hole.

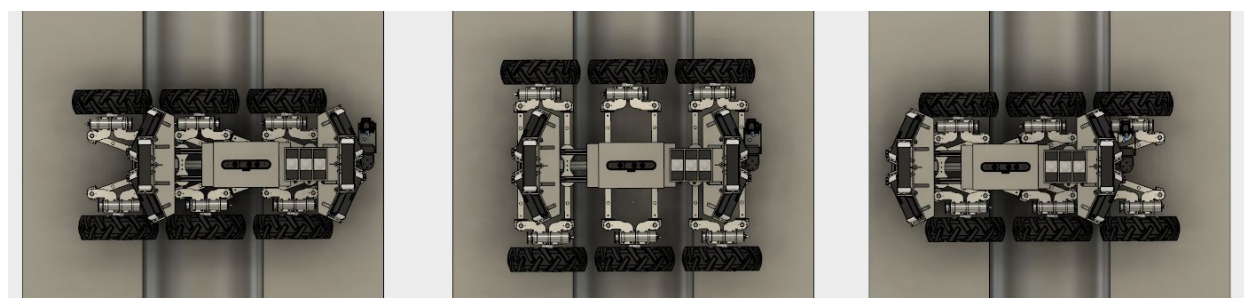


Figure 4.5. SIAR robot changing the center of mass while crossing the gutter.

With the possibility to change the width it will be possible to perform a safe navigate inside different galleries types. Being the robot able to adjust its width to the gallery and gutter width. Figures 4.6 and 4.7 show the robot running in different types of galleries.

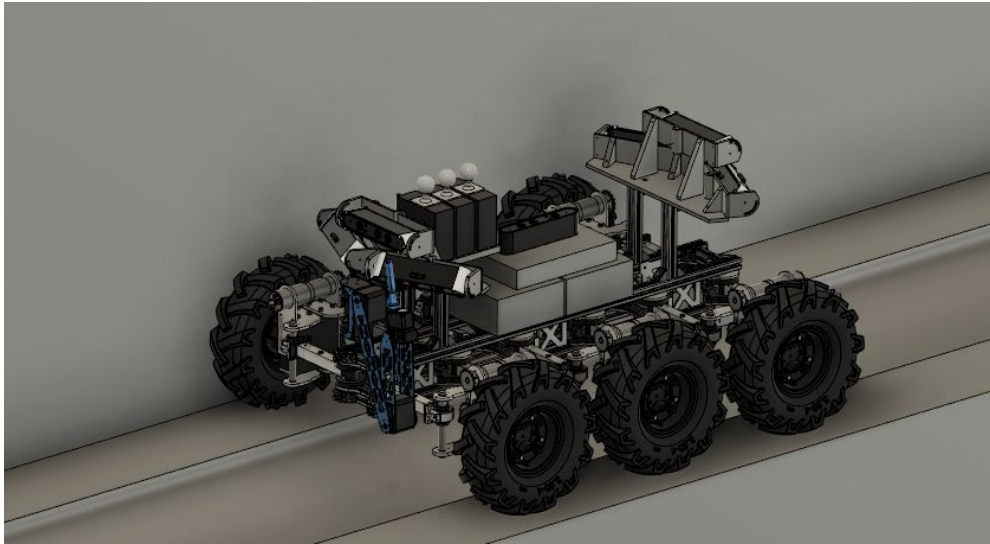


Figure 4.6. SIAR robot running over the gutter with the maximum width.

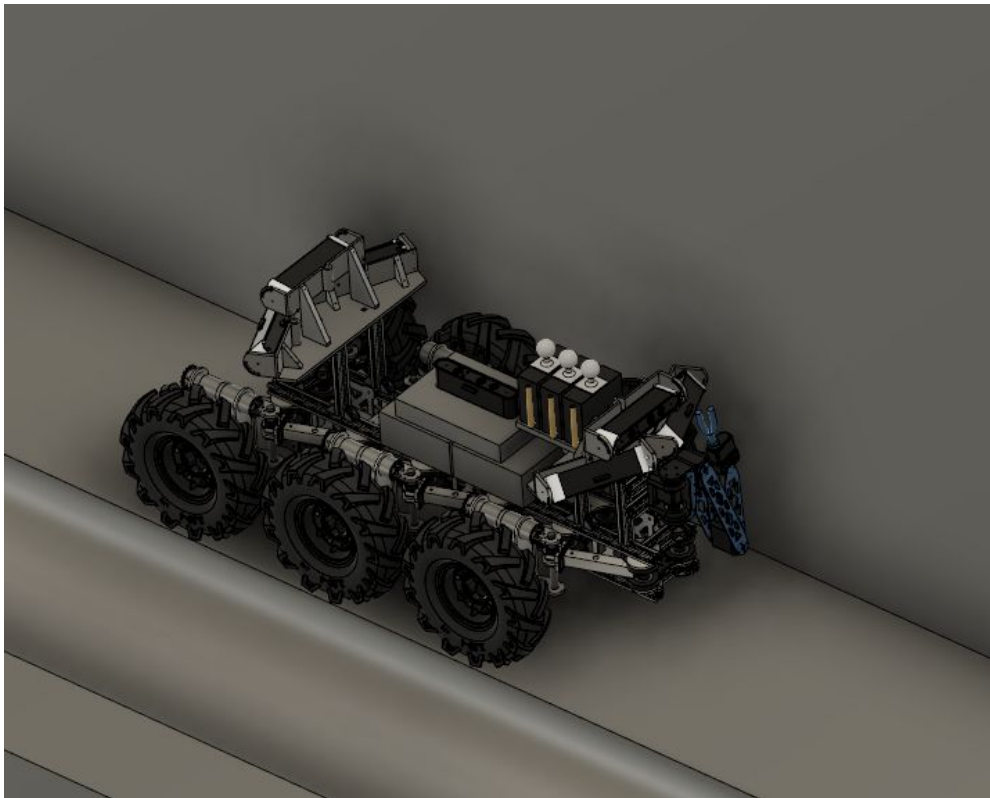


Figure 4.7. SIAR robot running on the side of the gutter in his minimum width.

The rotation over the gutter will be easier. Figure 4.8 shows a 90 degree rotation over the gutter.

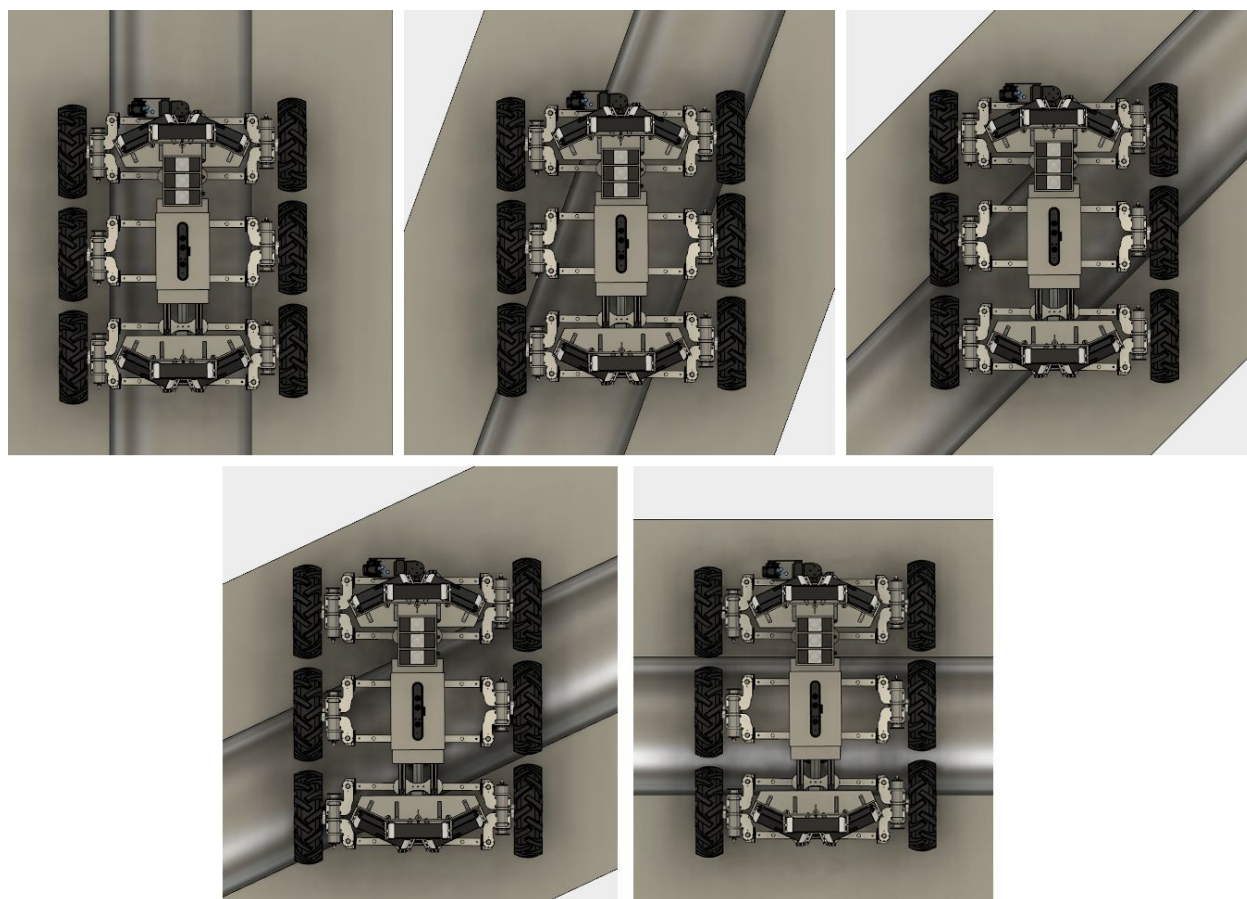


Figure 4.8. 90 degree rotation over the gutter.

Using this configuration it is possible to see that the robot will maintain at least 3 wheels touching the floor while rotating and most of the time it will have 4 wheels.

Another problem that uses the capability to change the width is when there is the need to change the direction in a detour. In this case the robot needs to adapt the width in a way that it will not touch the gallery walls. Figure 4.9 shows the robot changing direction in a fork.

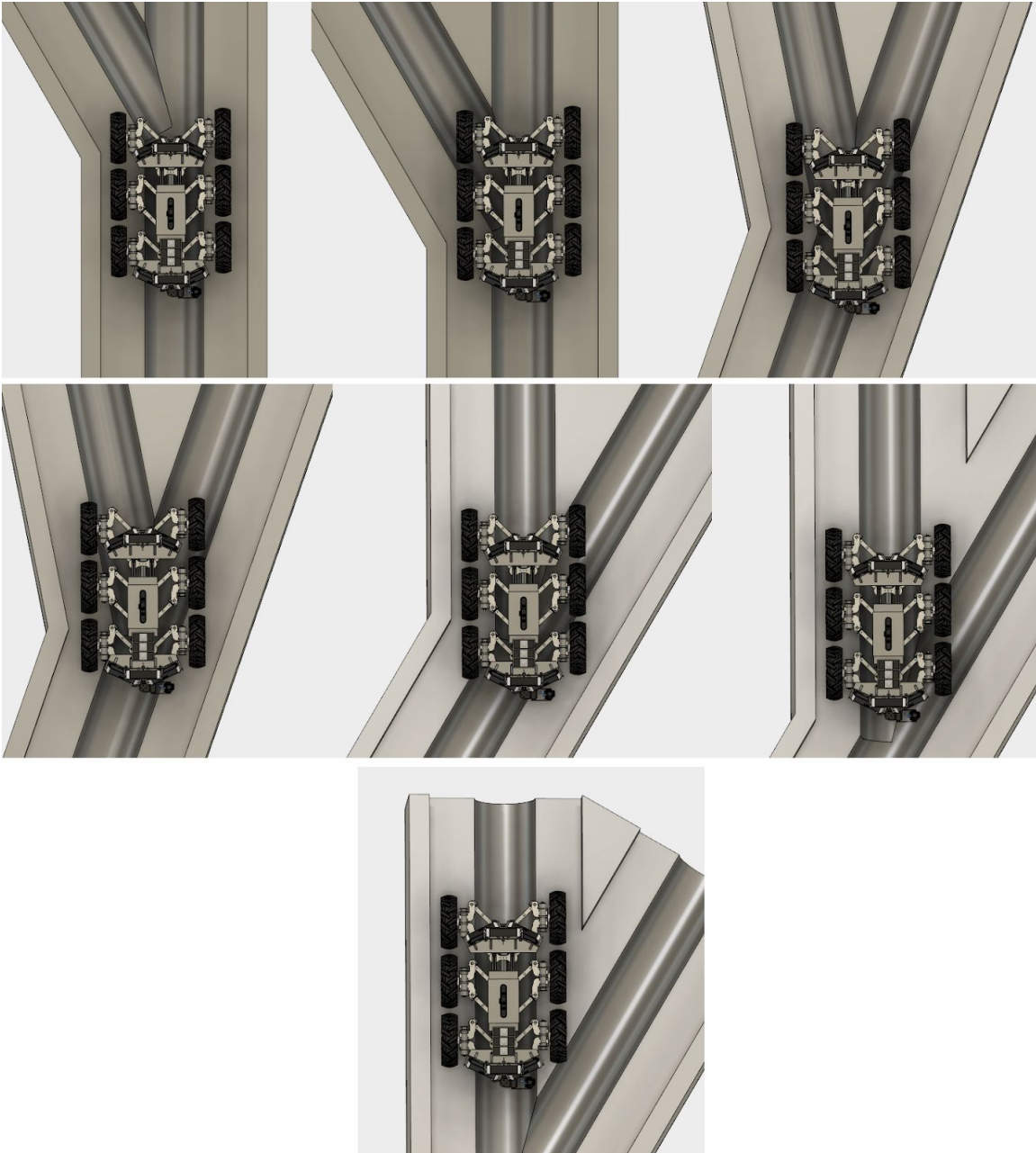


Figure 4.9. Changing direction in a fork

4.2 Inspection Capabilities

According to the Challenge document [ECHORD++, 2014], the following inspection capabilities are envisaged:

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

In this section the improvements of the inspection capabilities over the state presented in D28.1, Section 6 are presented. In particular, a method for the generation of a globally stable map, obtained from a complete experiment and an automatic system for detecting manholes in the sewer system, are presented in Sections 4.2.2 and 4.2.3, respectively.

4.2.1 Local 3D reconstruction

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

The 3D 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, but sometimes is also needed the third-person perspective in order to fully understand the robot with respect the scene.

In D28.1 a system for local 3D reconstruction was described. This system was demonstrated in real-time during the Phase I evaluation tests.

Figure 4.10 shows a snapshot of the information presented to the operator, we can see that 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.

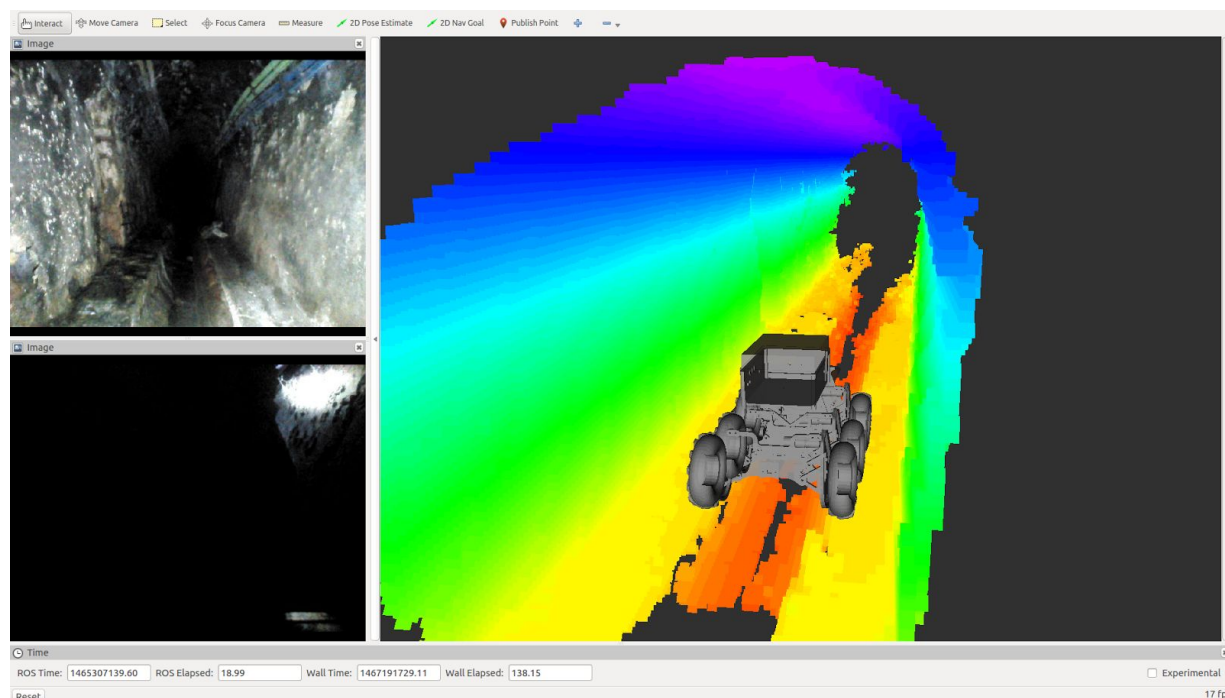


Figure 4.10. Different views of the operator while command the SIAR platform.
Left: First-person perspective (front and rear cameras). Right: Third-person perspective

4.2.2. Global 3D reconstruction

We have extended the procedure for obtaining Local 3D reconstruction of an area. This procedure was used to generate globally stable reconstructions, by integrating depth measurements from a complete experiment and correcting the odometry errors taking into account the manholes automatically detected.

The system requirements for 3D reconstruction for the robot operation are covered by the local 3D reconstruction. This will provide a good reconstruction of the environment, allowing to recognize possible defects or just for situational awareness of the operator. The SIAR approach for global 3D reconstruction is based on the application of the off-line robot trajectory optimization. It will consider the information provided by the sewer elements automatically detected (presented in next section) by the robot.

Thus, the approach will build a robot pose graph based on the motion provided by the visual odometry system, developed under the SIAR project. 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 4.11 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.

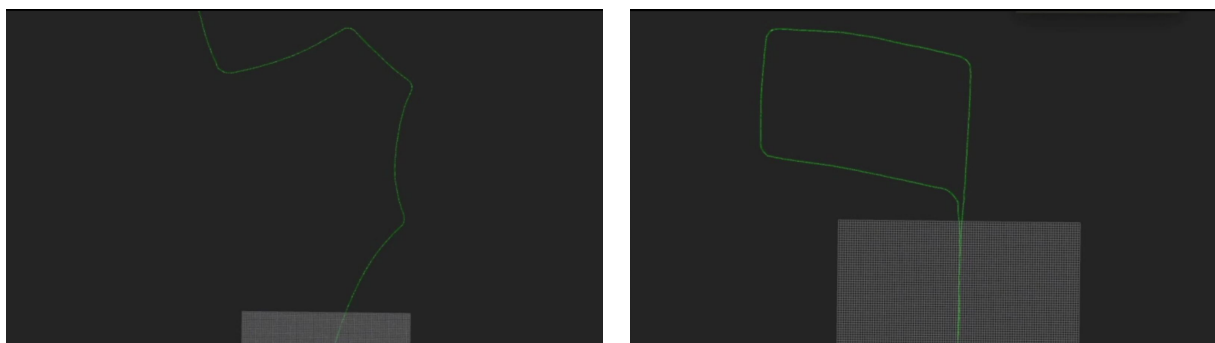


Figure 4.11. Optimization process over a real trajectory (400 meter long approximately). (Left) Original trajectory computed based on visual odometry. (Right) Obtained trajectory after non-linear optimization using manholes to detect close-loops

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 [Grisetti et al., 2010]. However, this approach goes one step forward and 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.

Given the sensor point-cloud pc_i at pose x_i and the point-cloud pc_j and pose x_j in a loop-closing, the scan-matching process establishes the transform that best aligns both point-clouds. In the literature, this transform is typically used as a constraint between both poses and its associated information matrix allows tuning the importance of such constraint into the nonlinear optimization process [Grisetti et al., 2010]. Instead, we propose including the alignment error into the optimization process, as we aim to build an accurate 3D map.

To this end, we transform each point-cloud to the global reference frame according to its associated pose pc_i^g and compute the alignment error between point-clouds as the averaged Euclidean distance between their individual 3D points. Then, we proceed to optimize the following expression:

$$\arg \min_{\{X\}} \left[\sum_{i=1}^N \left(E_i + \sum_{l=1}^{P_i} D(\mathbf{pc}_i^g, \mathbf{pc}_l^g) \right) \right]$$

where E_i stands for the error between pose x_i and x_j according to the odometry information and the function $D(\mathbf{pc}_i^g, \mathbf{pc}_l^g)$ computes the average Euclidean distance between the given point-clouds in the global frame. P_i is the number of loop-closures that affect pose i .

The computation of $D(\dots)$ could have a significant cost if the point-clouds involved are large because it needs to compute the match between the 3D points of both clouds each time, slowing down the optimization process. However, assuming the poses to be optimized are not far from the final estimates, thanks to the previous step, we can pre-compute the 3D point data association step between point-clouds. This way, function $D(\dots)$ only needs to recompute the average distance between 3D points because the association is known.

After the pose optimization stage the point-clouds of every single pose can be projected into world reference frame in order to build a globally consistent map of the robot trajectory. Figure 4.12 shows a 3D reconstruction of the sewer during the trajectory presented in Fig. 4.11. It can be seen in Fig. 4.12 how the general structure of the sewer section can be visualized. This 3D reconstruction is metric, so that the operator can take measurements of the different areas of interest in the environment.



Figure 4.12. A section of the 3D reconstruction of the sewer

4.2.3. Automatic Detection of Sewer Elements

An additional functionality is the localization of sewer elements, like manholes, inlets, crossings and others. In D28.1 we showed how the reconstructed 3D clouds give information about those elements.

Detecting the elements can be based on a geometric analysis of such data, 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.

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 4.13). 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 4.13. 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 4.14. The input images are downsampled to 80x60, as the manholes should remain detectable. The network is composed by four convolutional layers followed by ReLU activations and MaxPooling, except for the last one. We can see how we reduce the size of the inputs until having a tractable amount of information. The output of the fourth convolution layer is connected to a fully-connected layer of 100 neurons and ReLU activations. Finally, the output layer is composed by a single fully-connected neuron with sigmoid activation functions. This CNN has roughly 9000 parameters and its computation time is below 1.5ms in a regular i7 laptop.

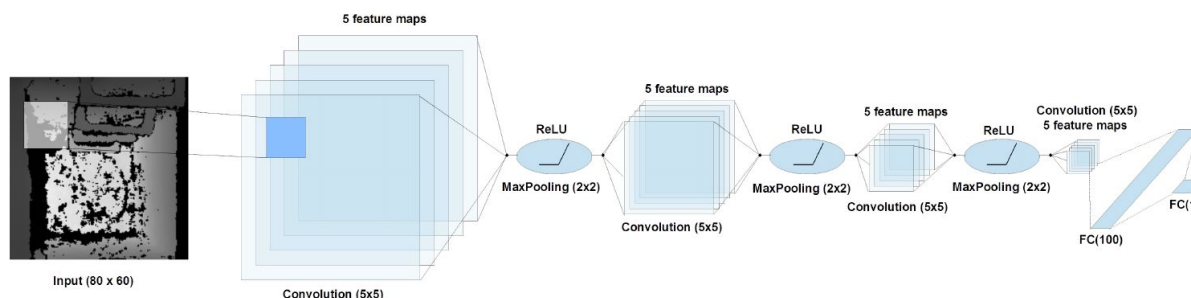


Fig 4.14. Convolutional Neural Network for automatic manhole detection based on depth images

The CNN has been trained and validated with a dataset 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 validate the CNN and are not included into training process.

We implemented the networks using Python and the Keras library [Cholet, 2015]. The learning algorithm used to train the networks is the Adaptive Moment Estimation (Adam), an algorithm for first-order gradient-based optimization.

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 2 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 2. Manhole detector confusion matrix

4.3 Robot localization in the sewer system

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 4.15). This system has been accepted for publication as a part of the International Conference on Intelligent Robots and Systems, 2017 [Alejo et al., 2017].

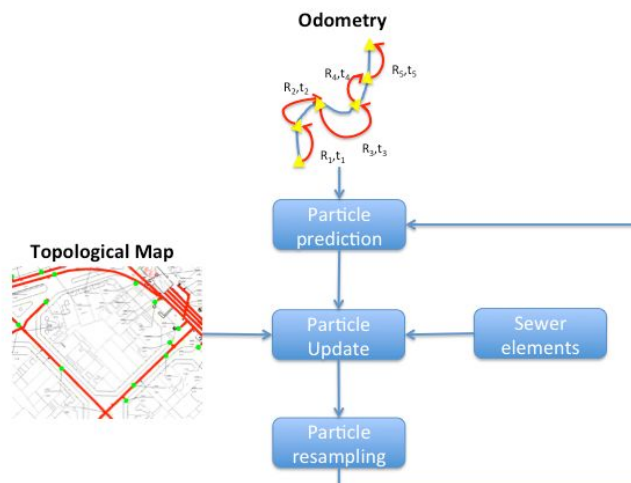


Figure 4.15. Topological map based localization approach. The detected sewer elements, like manholes, and a map of the sewer network including such elements are used to provide a global localization tracking within this map.

- 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 [1] developed for stereo-vision and adapted to RGBD. This block takes as input the RGB and depth flows of the frontal camera of the robot, matches robust features from consecutive or closely spaced frames and then obtains the relative pose between cameras that minimizes the re-projection error. The reader is referred to the D28.1 document, Section 5.1, in which the details on the RGB-D based Odometry can be found.
- Manhole detector.** This module 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 4.3.1).

4.3.1. 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 4.16 shows the topological map used in the experiments from which a graph that will be internally used by the localization module is extracted. This graph will contain 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.

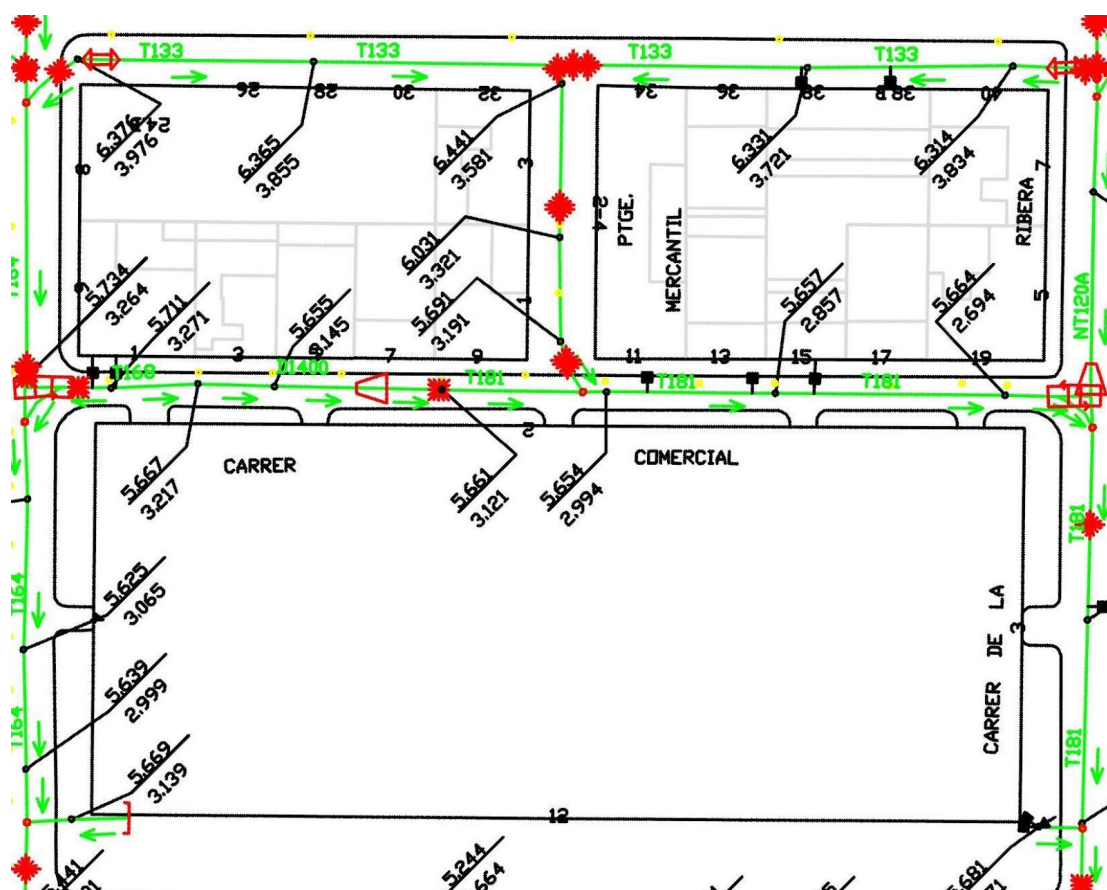


Figure 4.16. Topological map used as input for generating the internal graph in the localization module.

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.

4.3.2 Field experiments

Two different experiments (Experiment 1 and Experiment 2) were carried out in the sewers near the Mercat del Born, Barcelona on the 17th of May. Figure 4.17 represents the path followed in each experiment as well as the internal topological map stored in the localization module. Each node is identified with a different number and classified as a manhole (represented with a white circle) or fork (with a purple mark). During these experiments, the robot was teleoperated from a short distance.

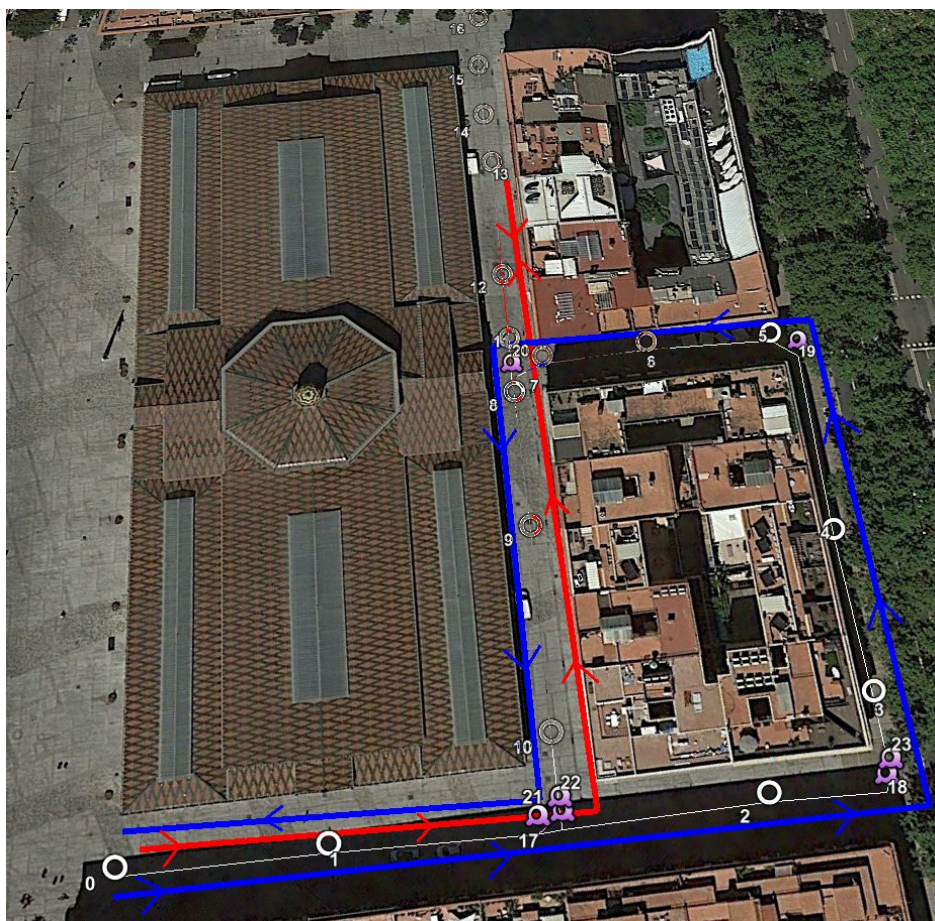


Figure 4.17. Scenario where the experiments were carried out. The blue and red paths are followed in Experiment 1 and 2, respectively. Manholes are labeled in a white circle while forks are labeled in purple.

The complete localization system has been successfully tested with the data gathered in the sewers. As the algorithm is randomized, the results presented here are calculated taking into account the output from 30 different executions, all of them using the same input data from both Experiment 1 and 2. The total

traversed distance in these experiments was about 400m in both cases. Figure 4.18 shows an example of the trajectories of the robot in Experiments 1 & 2, as estimated in an execution of the proposed localization algorithm. It is worth to note that the obtained trajectories are smooth and strictly follow the topological map.

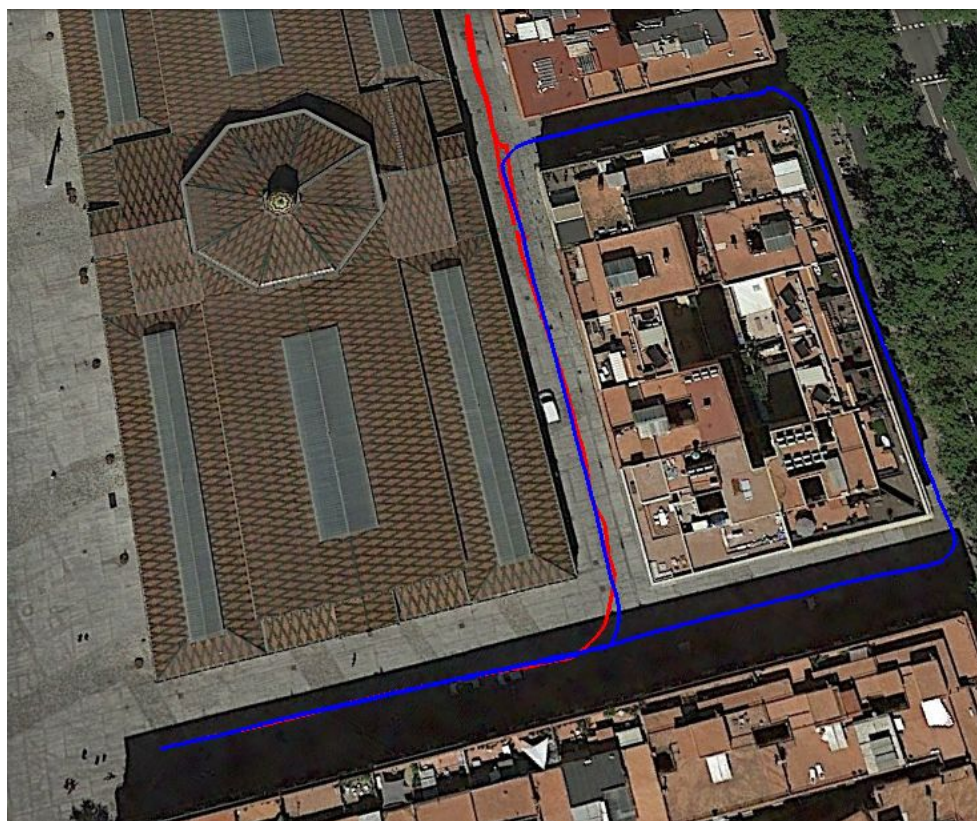


Figure 4.18. Localization results of the robot's position in Experiments 1 (blue) & 2 (red).

Figure 4.19 represents the distribution of the distances of the estimated position of the robot vs the ground truth position of the manhole above the robot, in each execution of the algorithm in Experiment 1. In this experiment, the robot follows a path with a loop and passes below 11 manholes, revisiting only one of them (the first and tenth manhole). First of all, the main result is that the localization algorithm succeeded to perform an accurate localization of the robot in all of the executions. Moreover, results show that the median distances of the obtained particle sets to the actual position of the manhole only exceeded $1m$ at the time the robot was under the manhole 7.

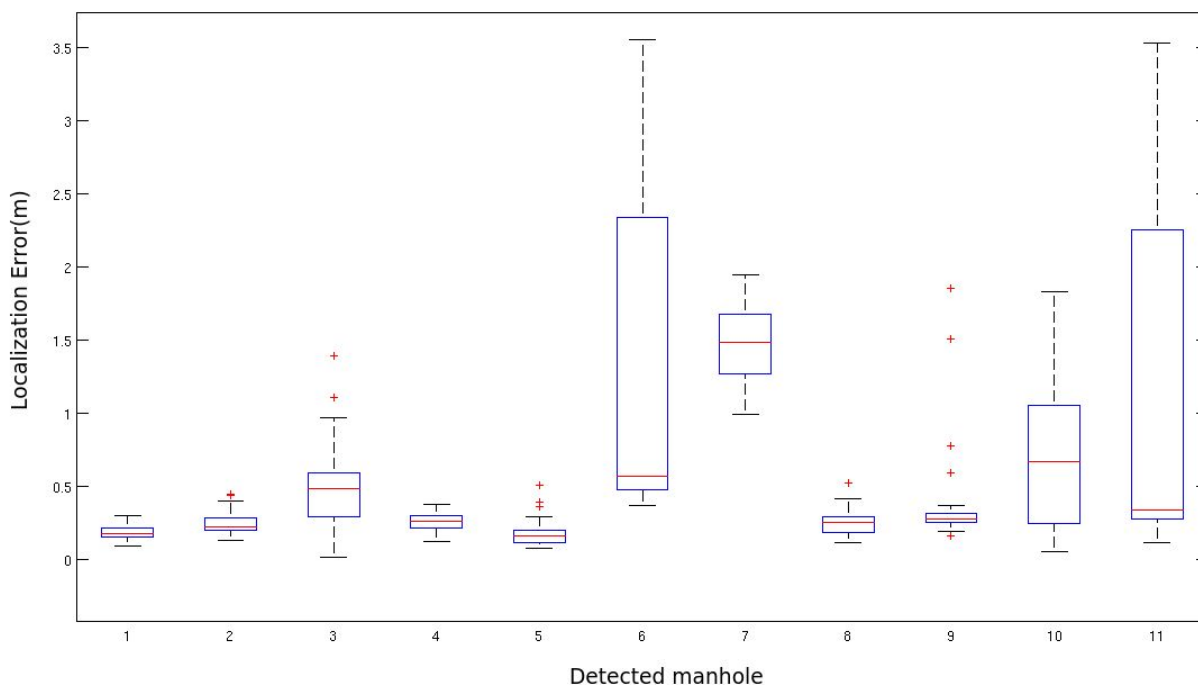


Figure 4.19. Distribution of the distances of the estimated position to the ground truth position of each manhole in Experiment 1.

Figure 4.20 represents the same information as the previous figure but plotting the results obtained in Experiment 2. In this experiment, the robot follows an 'L' shaped path before being commanded to turn 180 degrees and go back. This kind of maneuvers are difficult to correctly predict with odometry measures and thus the localization results are noticeably worse in this experiment starting from manhole 5. However, the median results are usually in the surroundings of $1m$ of localization error, and they rarely exceed $3m$.

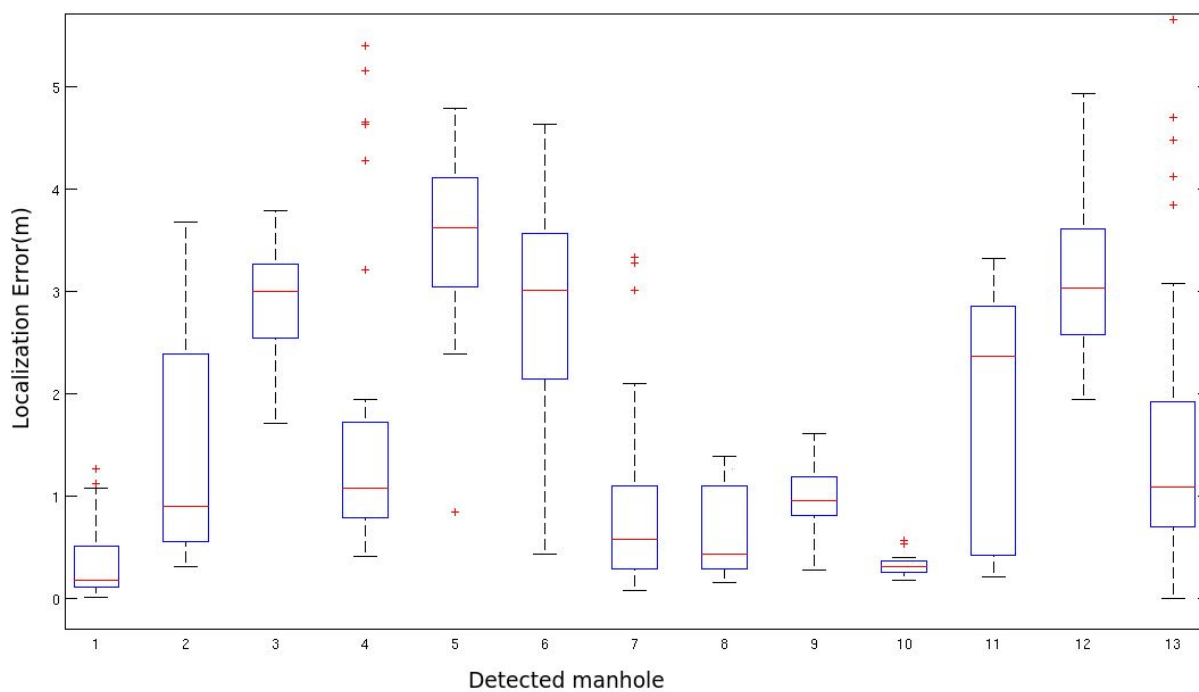


Figure 4.20. Distribution of the distances of the estimated position to the ground truth position of each manhole in Experiment 2.

References

- [Alejo et al., 2017] D. Alejo, F. Caballero and L. Merino. “RGBD-based Robot Localization in Sewer Networks”. To appear in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, September 2017.
- [Chollet, 2015] F. Chollet, “Keras, Deep Learning library for Python” <https://github.com/fchollet/keras>, 2015. Accessed on June, 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
- [Grisetti et al., 2010] G. Grisetti, R. Kuemmerle, C. Stachniss, and W. Burgard, “A tutorial on graph-based SLAM,” *Intelligent Transportation Systems Magazine*, IEEE, vol. 2, no. 4, pp. 31–43, 2010.
- [Jaillet et al., 2008] L. Jaillet, J. Cortés, T. Siméon. “Sampling-based path planning on configuration-space costmaps”. *IEEE Transactions on Robotics* 26 (4), 635-646, 2014.
- [Kuffner et al., 2000] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, San Francisco, CA, 2000, pp. 995-1001 vol.2.
- [Rivera et al., 2013] Rivera, N., Baier, J. A., & Hernández, C. (2013, May). Weighted real-time heuristic search. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* (pp. 579-586). International Foundation for Autonomous Agents and Multiagent Systems, 2013.
- [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.