

# 3D Localization for the MagneBike Inspection Robot

---

Fabien Tâche, François Pomerleau, Gilles Caprari, Roland Siegwart

Autonomous Systems Lab

Eidgenössische Technische Hochschule Zürich (ETHZ)

8092 Zürich, Switzerland

{ftache, pomerlef, caprarig, rsiegwart}@ethz.ch

Michael Bosse

CSIRO ICT Centre

Autonomous Systems Laboratory

4069 Kenmore, Australia

mike.bosse@csiro.au

Roland Moser

R&D Inspection Technologies

ALSTOM Power Service

5401 Baden, Switzerland

roland.moser@power.alstom.com

## Abstract

The MagneBike inspection robot is a climbing robot equipped with magnetic wheels. The robot is designed to drive on 3D complexly shaped pipe structures, therefore it is necessary to provide 3D visualization tools for the user who remotely controls the robot out of sight. The localization system is required to provide a 3D map of the unknown environment and the 3D location of the robot in the environment's map. The localization strategy proposed in this paper consists of combining 3D odometry with 3D scan registration. The odometry model is based on wheel encoders and a three axis accelerometer. Odometry enables the tracking of the robot trajectory between consecutive 3D scans and is used as a prior for the scan matching algorithm. The 3D scan registration facilitates the construction of a 3D map of the environment and refines the robot position computed with odometry.

This paper describes in detail the implementation of the localization concept. It presents the lightweight small sized 3D range finder that has been developed for the MagneBike. It also proposes an innovative 3D odometry model which estimates the local surface curvature to compensate for the absence of angular velocity inputs. The different tools are characterized in detail based upon laboratory and field experiments. They show that the localization concepts reliably track the robot moving in the specific application environment. We also describe various techniques to optimize the 3D scanning process, which is time consuming, and to compensate for the identified limitations. These techniques are useful inputs for the future automatization of the robot's control and optimization of its localization process.

## 1 Introduction

The goal of the MagneBike project, started in 2006, is to develop a mobile robotic platform capable of inspecting power plant facilities, focusing on the inner casing of steam chests for a specific case study. Due to size constraints and the complex shape of the environment, the first challenge was to develop a compact system with climbing and obstacle passing capabilities. In this framework, the MagneBike locomotion concept, shown in Figure 1, was designed. The robot consists of two aligned magnetic wheels units with integrated lateral lever arms. These arms have two complementary functions: they can be used to slightly lift off the wheel in order to locally decrease the magnetic attraction force when passing concave edges or to laterally stabilize the robot when gravity is unfavorable. Steering is ensured by an active degree of freedom (DoF) on the front wheel and surface adaptation is provided by a free joint on the fork. This locomotion concept has a very high mobility and enables driving on complex 3D industrial environments that are not necessarily designed for robots. The robot can climb vertical walls, follow circumferential paths inside pipe structures and can also pass over complex combinations of convex and concave step obstacles with almost any inclination regarding gravity. It requires only limited space to maneuver because turning on spot around the rear wheel is possible. The compact and lightweight climbing robot ( $185 \times 131 \times 153 \text{ mm}^3$ <sup>1</sup>, 3.5 kg) and its high mobility locomotion concept are presented and analyzed in detail by (Tâche et al., 2009).

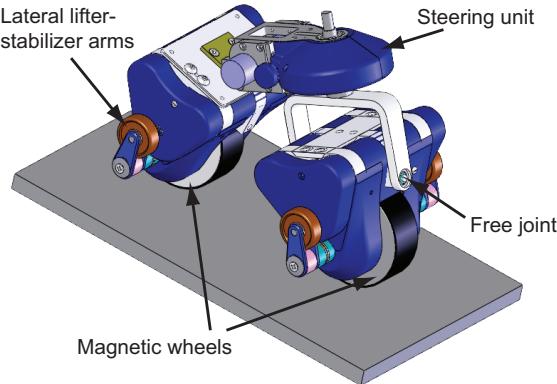


Figure 1: MagneBike locomotion concept: CAD model.

With the locomotion problem effectively solved, the next challenge consists in defining a localization and control strategy. Because the robot is mainly driven in confined environments out of the user's field of view, the goal is to provide a 3D visualization tool/feedback that helps the human remote controlling the robot to make correct control decisions. This means providing the 3D position and orientation of the robot (3D localization) in a 3D visualization of the environment (3D mapping). Moreover, localization is necessary to precisely locate any defects identified during inspection.

3D localization and mapping are actually intermediate steps towards an implementation of semi-autonomous control, where the robot would automatically plan its trajectory towards

<sup>1</sup>Values may differ from (Tâche et al., 2009), because the robot embeds a new steering unit that is compacter and more reliable.

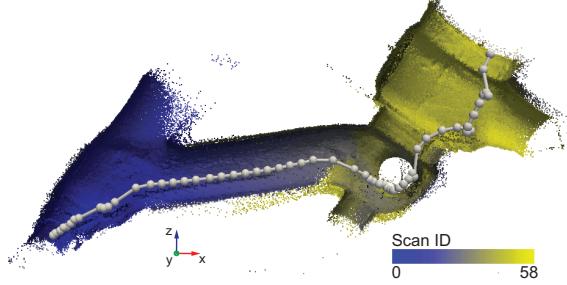


Figure 2: 3D point cloud reconstruction (cut view) of the inner surface of the real application environment. Grey dots indicate the positions (3D range finder frame) where 3D scans were taken.

a goal position or follow an inspection scenario defined by the human operator. In this control mode, the user would also be responsible to take high level decisions (*e.g.* activation of the lifting mechanism) based on the robot recommendations and/or using the 3D visual feedback, respectively the 3D scans as well as the localization of the robot in the map. Semi-autonomous control would actually require a non-trivial solution to the path planning/following problem due to the complex shape of the 3D environment and the constraints on how to autonomously pass over obstacles with the active system. Semi-autonomous control is therefore set as long term objective of this work and the experiments presented in this paper were performed in remote control mode. The robot however embeds a single board computer that controls local tasks involving embedded sensors and direct interactions with the environment or predefined movements. The MagneBike for instance controls behaviors such as the deformation control that avoids robot deformations when driving on irregular surfaces and the control of the stabilizer arms for lateral stabilization (refer to (Tâche et al., 2009) for details). The robot is tethered with a cable that is used for power supply, Ethernet communication and security. A cable is in any case required because wireless communication might cause problems in thick metallic structures and power plant owners insist on having a security cable to pull the robot back in case of failure.

This paper then focuses on the 3D localization and mapping strategy, its implementation and its detailed characterization through field experiments. It first introduces the constraints and challenges of the localization problem in Section 2. The 3D localization strategy - the combination of 3D odometry, 3D scanning and matching - is also discussed in this section. Section 3 discusses the choice of the 3D range finder and its optimal implementation on the robot. The algorithms and tools used to solve the localization problem are described in Section 4. They are then characterized by presenting results of laboratory and field experiments in Section 5. Finally, conclusions and future work are given in the last two sections.

## 2 Localization for the MagneBike: problem statement and approach

The MagneBike application implies 3D localization in a new type of environment for mobile robots. In this section, we then describe the specifics of this environment and the required accuracy of the localization system. Once the constraints are presented, related work is discussed and a localization strategy is proposed.

### 2.1 The MagneBike's environment: specificity and requirements

The environment specifications have already been presented in a former paper (Tâche et al., 2009), primarily focusing on requirements and constraints for the locomotion system. Typical environments (Figure 3) are analyzed again, but this time from the point of view of the localization problem. The following constraints are important, because they differ from the conventional indoor and outdoor robotic applications:

- a. The robot drives in a confined space. It is then important to provide a 3D visualization tool to the user who cannot see the robot at all, once it entered the structure.
- b. The rather small size of the structure (radii from 200 to 700mm) limits the range of observable features.
- c. The structure is built out of metal. The characteristics of this material (brightness, texture, roughness) can influence the performance of range finders (more details are given in Section 3).
- d. The inner casing surface is also rather homogeneous. Color and texture features cannot be used as robust environment descriptors. Furthermore, there is no natural light in the closed environment. Thus, the robot has to bring its own light source in order to use vision sensors.
- e. The environment does not contain the typical 2D geometric features (such as straight lines or planes) and geometric constraints (perpendicularity of lines and/or planes) often used as descriptors for indoor mobile robotics. However the environment can be described by parts of cylinders, spheres or circles (edges at junctions or diameter changes).
- f. The shape of the environment is usually known by the inspector, but an exact 3D model usually does not exist. A model could be reconstructed from 2D drawings, but because the installations to inspect are rather old, it is never ensured that these drawings are still available or up-to-date (*e.g.* missing hole obstacles that could block the robot). The CAD models presented in this paper are for instance rough reconstructions (without details, *e.g.* small holes, ridges) based on existing 2D drawings. For generality purpose, it is later assumed that a 3D model is not available.
- g. In order to control the locomotion system (automatically or not), it is important to generate a model with a sufficient resolution that enables the detection of obstacles which must be avoided or overcome by the robot.
- h. Finally, because the MagneBike is a climbing robot that can drive on surfaces with any inclination regarding gravity, the localization requires to solve a full 3D problem,

respectively six DoF. The problem cannot be compared to most of indoor/outdoor robotics applications for which the movement is often limited to horizontal planes (2D) or with rough terrain exploration rovers which consider slight orientation changes around the horizontal plane (Bonnifait and Garcia, 1999) and small vertical displacement when negotiating obstacles (Lamon, 2005).

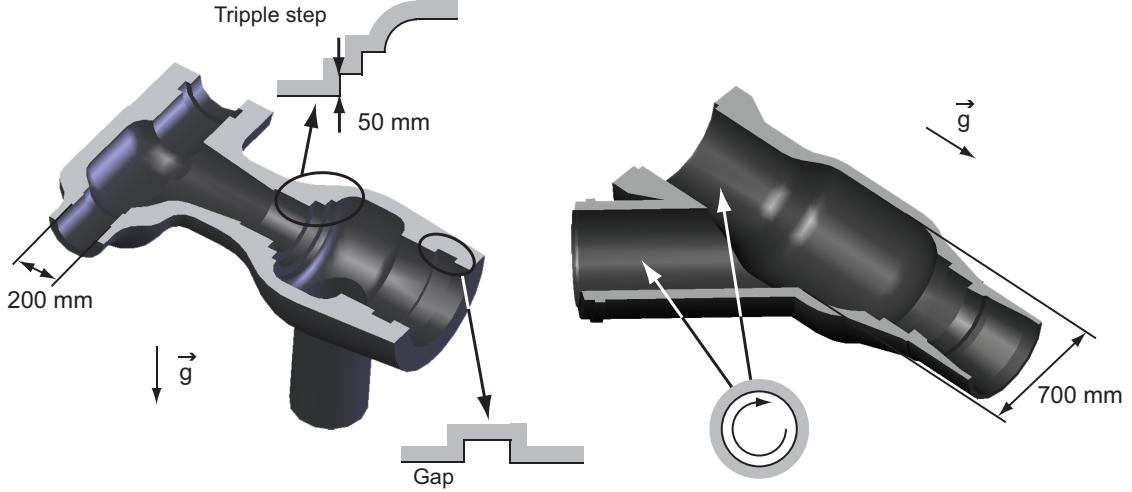


Figure 3: CAD model of typical environments for the MagneBike.

Regarding the accuracy of the localization and mapping system, the following objectives were defined:

- For safe remote operation, the robot localization system should provide a position feedback to the user with an accuracy of a few centimeters ( $\pm 5\text{cm}$ ).
- The inspection task however requires to locate identified defects with more accuracy, respectively in the centimeter range ( $\pm 1\text{cm}$ ).
- Regarding the orientation, the accuracy should be of  $\pm 10^\circ$ , in order to ensure to drive the robot over obstacles with a safe angle of attack, especially on convex edges as demonstrated by (Tâche et al., 2009).
- The map resolution should be high enough to detect obstacles/holes with a size of 1cm at a distance of 0.5m. This allows the user to plan an appropriate trajectory to avoid/overcome obstacles with the nonholonomic robot.

## 2.2 Related work

As listed in the previous section, the inspection robot environment differs from typical indoor/outdoor mobile robots worlds. The related work discussion therefore focuses on localization strategies for robots dedicated to similar environments or having similar constraints.

For remote controlled robots, vision sensors embedded on the robot provide the necessary feedback to the remote user. Camera image feedback is sufficient to control the robots at pipe junctions, to guide the robot on curves, to avoid obstacles, to detect obstructions or

the end of the area to inspect. The camera is often also the inspection sensor. Odometry is simply based on wheel/caterpillar encoders. The length of the tether (if any) can also be used to get a good approximation of the overall distance traveled. Sewer and petrochemical pipe inspection robots (Inuktun, 2009) and the Toshiba micro pipe inspection robot (Suzumori et al., 1999) are typical examples.

When some autonomy is required, the robots use the environment features as landmarks for localization and navigation. For instance the sewer inspection robot *KANTARO* (Ahrary et al., 2007), a wheeled robot that passively adapts its wheel orientation to the bends in the pipe, combines stereo vision with a 2D laser scanner to detect landmarks. (Ahrary, 2009) also proposes a landmark detection technique based on a camera and infrared sensors. The landmarks such as manholes, pipe joints or inlets can be matched with an existing map of the environment. With this strategy, the robot achieves full autonomous localization and navigation. A similar strategy is used by (Hertzberg and Kirchner, 1996) to generate topological maps with nodes at the intersection of the pipe junctions. Nodes are perceived with a rotating ultrasonic transducer and classified using a neural network.

If the environment model is not available and the generation of a detailed map is required, it is necessary to embed 3D mapping sensors. A typical approach is presented by Nüchter (Nuechter et al., 2003) for the *KURT2* robot, a wheeled robot designed for sewer pipe inspection. The robot is equipped with a 3D laser scanner, a rotating 2D SICK laser range finder (Ye and Borenstein, 2002). This sensor captures 3D point cloud measurements of the surrounding environment. The challenge then consists in constructing a consistent 3D model based on consecutive 3D scans. An iteration of this process basically consists of: acquiring a new 3D scan, aligning it to the existing data based on odometry using a scan matching algorithm such as Iterative Closest Point (ICP) (Zhang, 1994), and improving alignment until reaching a cost function minimum. Finally, the new data is included in the existing map, which enables the reconstruction of the sewerage pipe system.

Finally, the most advanced results in the field of inspection and exploration are the *Groundhog* and *Cave Crawler* robots, which are designed to explore abandoned mines (Thrun et al., 2004), (Morris et al., 2006). The *Groundhog* robot is fully autonomous and uses software packages which solve the SLAM problem: the robot builds a map and simultaneously localize itself relatively to this map. In addition to using 2D scan matching to build a 2D map of the environment, the robot also acquires 3D range scans at periodic intervals. These scans are used to analyze the free space available in front of the robot, in order to upgrade the 2D map to a 2.5D map which integrates the traversability of the local area. This information is finally used to efficiently plan the robot path.

To conclude this overview, the literature does report on systems with advanced 3D localization and mapping ability; however, these systems are either too large and heavy or/and have only limited climbing mobility. The aforementioned robots are for instance designed to work in horizontal pipes/corridors networks with low elevation changes and they are only able to pass over small obstacles. On the other hand, the climbing robot literature reports on many small sized climbing systems with high mobility, but not integrating localization and mapping sensors. Papers actually mainly focus on locomotion concepts; refer to (Tâche et al., 2009) for related work about locomotion concepts and mobility. In this paper, we however

present a small sized climbing robot with a high climbing mobility that embeds sensors for 3D localization and mapping in complex shaped environments from which no map is known ahead of time. Furthermore, the robot is driving in full 3D environments, with the ability to reach any orientation with respect to gravity.

### 2.3 Localization: strategy

Because the MagneBike is a climbing robot, solving the localization problem consists in determining its position and orientation (six DoF) in a 3D space. The number of reachable positions and relative orientations are however limited, since the robot is constrained to drive on the environment's surface. Before presenting the localization strategy that is selected for the MagneBike, other approaches and their rejection criteria are briefly discussed.

Since the robot is working in thick metallic structures located in indoor environments, using a global positioning system (GPS) or a beacon system is not an option. Tracking systems are also not suitable, because there is no location from which the system can continuously track the robot in the closed environment composed of L and T intersections. Approaches based on inertial measurements (IMU) are also rejected due to their unbounded error growth. These kinds of sensors (*e.g.* gyroscopes) are then not appropriate for the MagneBike that is moving slowly in its maneuvers about the complex environment. Furthermore magnetometers are of no help in the specified metallic environments.

Odometry approaches are better suited, because their errors drift with distance and not with time. An approach based on a full complex internal state model is however also discarded, as it would necessitate the integration of many sensors to precisely measure the robot state in space: for instance sensors to measure the orientation of the free joint on the front wheel unit fork and the contact angles between the wheels and the environment surface (refer to Section 4 for a detailed robot model). The contact angles could also be computed by projecting a 3D model of the robot on a detailed model of the environment and applying physical contact constraints. This is however a very complex geometrical problem and the environment model is assumed to be unknown anyway, as discussed in the specifications.

Hence, we propose a localization strategy based on a simplified 3D odometry model which uses only wheel encoders and a three axis accelerometer to estimate the surface orientation with respect to gravity. Since the robot moves slowly, it can be assumed that the accelerometer only measures the gravitational acceleration vector. Outliers of the acceleration values (*e.g.* due to shocks when the robot gets magnetically attracted by a step obstacle) are easily filtered. 3D odometry is then combined with 3D scan matching. Odometry is used to track the position between scans as well as prior knowledge for the scan matching algorithm. Starting from a known position, the strategy then consists in continuously performing the four following steps:

- a. **Move - 3D odometry:** the robot moves and tracks its position and orientation using 3D odometry.
- b. **Stop - 3D measurement:** the robot stops to get a high resolution 3D measurement of the environment.

- c. **Local map registration:** a scan matching algorithm is used to compute the 3D transformation between the two last 3D scans (taken at time  $t - 1$  and  $t$ ). The result of the odometry computed in step (a) is used as prior knowledge for the scan matching algorithm: this allows to update the robot position as well as to locally extend the 3D map.
- d. **Global registration:** if a pairwise matching (step c) were only performed between consecutive scans, the scan matching error would propagate along the robot trajectory. If redundant matching were to be performed (*e.g.* between non consecutive scans if the robot explores the same location from different points of view), a global optimization algorithm could be used to optimize the network of 3D transformations. This last step allows to decrease the error propagation and update the global map.

This approach is then very similar to the one of the *Groundhog* (Thrun et al., 2004), but extended to full 3D and applied to a new type of environment. In order to use this strategy, it is first necessary to find or design the appropriate 3D range finder for the MagneBike. This is presented in the next section, while the implementation and performance of each step of the localization strategy are detailed in Sections 4 and 5.

## 3 3D range finder for the MagneBike

### 3.1 3D range finder selection

The first issue is to select the appropriate 3D range finder for the MagneBike. It first has to fit on the small sized robot and satisfy payload constraints. Moreover, it has to work properly in the environment to inspect, which specific characteristics have been described in Section 2.1.

Because the environment surface (texture, color) is rather homogeneous, stereo vision is not a promising solution to build a 3D representation of the environment. The low amount of detected features that can be matched on both images and used as 3D measurements would be insufficient.

3D time-of-flight cameras that deliver 3D images at high frame rate, such as the *SwissRanger* (Weingarten et al., 2004) have two main limitations regarding this application. Preliminary tests showed that such sensors have very low precision when the incidence angle to the target is high, especially on metallic surfaces. Having such a sensor pointing forward on the MagneBike would lead to this worst case. Furthermore, even if the sensor provides a 3D image, it is still necessary to rotate it to get a full 360° local visualization of the environment because of its rather limited field of view.

A laser range finder which provides high densities point clouds and allows to construct 3D maps of the environment is the best option for the MagneBike application. The typical rotating SICK (Wisspeintner et al., 2008) or Velodyne (Beeson et al., 2008) sensors are very precise but far too big and heavy for the robot. Our choice then converged to the laser range finders from Hokuyo, especially the URG-04LX (Kawata et al., 2005) because of its small

size ( $50 \times 50 \times 70 \text{ mm}^3$ ) and light weight (approximately 0.16kg). It has however to be verified that this sensor works properly in the steam chest environment and can be upgraded into a 3D LADAR of reasonable size by adding a rotational DoF. These two aspects are addressed in the following sections.

### 3.2 Hokuyo-URG04LX characterization

Before using the Hokuyo URG-04LX laser range finder, we had to verify that it would be suitable for measuring the specific steam chest environment. In this framework, a detailed characterization was performed (Kneip et al., 2009). The former paper provides a detailed and complete characterization of the sensor by checking the commonly known issues with 2D laser range finders (Stone et al., 2004) and their dependencies on, for example, target surface properties, distance, and incidence angle. The usual experiments known from the works of Ye and Borenstein (Ye and Borenstein, 2002) on the SICK LMS200 and Alwan (Alwan et al., 2005) on the PBS-03JN have been extended with some specialized tests to check the dependency of the measurement on the sensor orientation or the influence of the ambient light. A characterization of the sensor was also done by the team of Borenstein and their results were actually published simultaneously (Okubo et al., 2009).

Here follows a list of the main results. For details, refer to (Kneip et al., 2009), the experiments done on the steam chest material provided similar characteristics to the electrical steel presented in the former paper:

- Drift effects: there is a continuous drop of the measured distance over the first 30 minutes, the sensor should then be started half an hour before the beginning of experiments for better performance.
- Sensor spatial orientation: the measurements are affected by the sensor orientation, especially when it is laying upside down. For instance, the relative error at 1.5m (-2.4%) is slightly higher than the relative error in the normal upstanding position (-1.7%).
- Ambient light: the tests show that the absence of ambient light (sensor operating in the closed steam chest environment) has no significant influence on the sensor readings.
- Target surface properties (color, brightness, material): the tests showed a clear influence of these three parameters on the relative measurement error. The standard deviation is for instance slightly higher for dark surfaces (3.4mm) than for bright surfaces (2.3mm), but the repeatability of the measurements is sufficient.
- Target distance: the most critical result is that certain types of targets are not measurable up to the maximum range (5.6m). Tests on the steam chest material show that the sensor can measure up to 5m if the incidence angle is low.
- Incidence angle on the target: the standard deviation is increasing with the incidence angle. The critical result is that the maximum incidence angle that still supports a reliable measurement is strongly depending on the target's shininess (reflectivity). In the steam chest environment, the maximal incidence angle at 1.5m is  $70^\circ$ . This effect

will of course affect the precision of the 3D model, because in pipe-type environments the incidence angle of the laser beam on the pipe surface will increase with distance when the robot drives longitudinally.

The Hokuyo is then the most appropriate sensor for the MagneBike. It is compact, lightweight and the detailed characterization did not point out any rejection criterion.

### 3.3 3D LADAR design and implementation

The common way to implement a 3D range finder based on a 2D scanner is to add an active DoF to provide the third dimension. For example, this was implemented with a 2D SICK laser (Surmann et al., 2001) and with a 2D Hokuyo sensor (Ueda et al., 2006). Due to size and weight constraints, an optimized custom design is necessary for the MagneBike. This section then presents the main design issues, the compact implementation as well as the characteristics of the 3D LADAR.

#### 3.3.1 Position and orientation of the rotation axis

The position and orientation of the rotation axis is an important design issue, because it influences the following 3D LADAR characteristics: the spatial density and coverage of the measurements, the scanning time and the space used by the sensor. These characteristics are discussed in detail for four possible ways of rotating the 2D scanning plane, respectively *rolling*, *pitching*, *yawing* and *yawing top* scan methods (Figure 4) according to Wulf's definition (Wulf and Wagner, 2003):

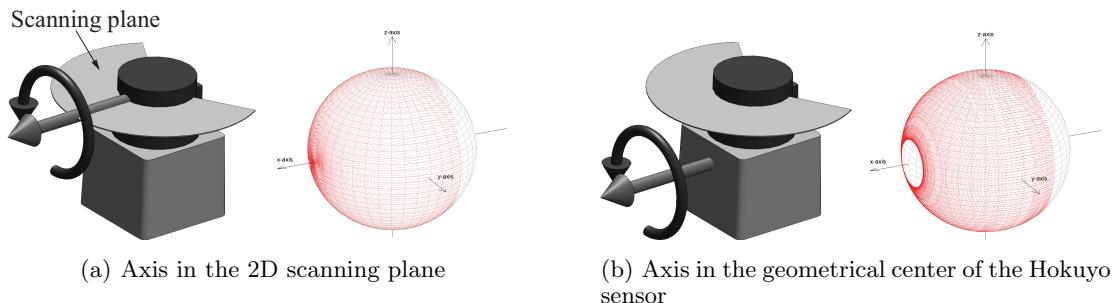


Figure 4: 3D scanner: position of the rotation axis. Projection of laser beams on a virtual sphere for the rolling scan configuration. If the rotation axis is not in the 2D scanning plane (b), a hole appears in the 3D scan. In any case, there is a blind zone due to the 240° field of view of the 2D range finder.

- *Rotation axis position:* if it is set in the geometrical center of the 2D range finder, the system requires less volume to achieve a 360° rotation. However it induces a hole in the 3D scan, as illustrated in Figure 4(b), if the axis is offset. In addition to blind zones, the border of these holes disturb feature detection algorithms since they are hard to differentiate from edges in the environment. Because of these two reasons we prefer solutions with the rotation axis within the 2D scanning plane [Figure 4(a)], if they can satisfy space requirements.

- *Spatial coverage*: the density of a point cloud is higher in regions that are close to the rotation axis as depicted in Figure 5. Due to the 240° field of view of the Hokuyo 2D range finder, it can be noticed that parts of these regions are scanned twice, thus increasing their point density. Because coverage is an important criterion for the registration algorithms (and for visualization as well), the coverage criterion is a strong argument against the rolling scan method that only covers the front field of view of the robot. The yaw scanning top method has a similar blind zone, but on the bottom side where the robot stands.
- *Spatial density*: the rolling scan method has the advantage to provide the highest density in front of the robot, providing the highest resolution in the driving direction. This is important for obstacle avoidance.
- *Scanning time*: the rolling and yawing top scans provide a full 3D scan by a rotation of only 180°. These methods have the advantage of being twice as fast as others.
- *Scanner size*: as discussed previously, the necessary space to get a 3D scan mainly depends on the position of the rotation axis. Because rolling and pitching scan methods require more vertical space (the most critical dimension of the robot), yaw and yaw top scan methods are more advantageous for the MagneBike.

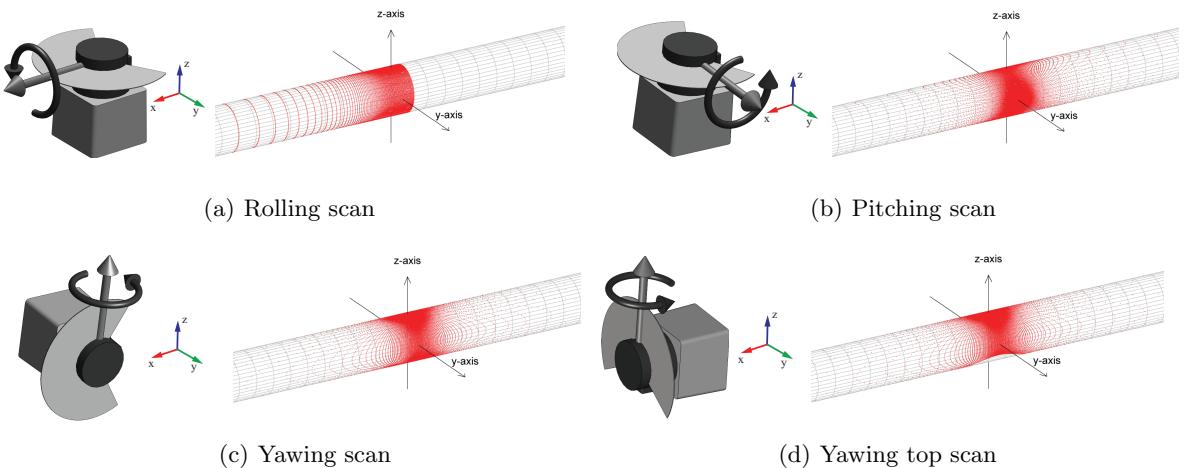


Figure 5: Comparison of possible scanning orientations as defined by (Wulf and Wagner, 2003): influence on the point density and surface coverage in a cylindrical environment. Blind zones (rolling and yawing top) are due to the 240° field of view of the Hokuyo sensor. The x-axis represents the robot driving direction.

To summarize rolling configuration is mainly rejected because it does not provide full 3D coverage. The pitching scan configuration, as the rolling configuration, has the main drawback to necessitate more vertical space and to be slower. Therefore the yawing top scan method is chosen, because it allows to scan faster than the yawing configuration that has similar characteristics.

### 3.3.2 3D LADAR implementation and characteristics

The optimal position and location of the rotation axis having been discussed, the final design choices and the 3D LADAR implementation are presented. The 3D range finder is actuated by a DC motor, this allows to set the scan resolution by adapting the motor speed. In order to get a compact design, a construction with worm gear transmission was used. Even if there is an encoder on the motor shaft, an absolute encoder is mounted on the Hokuyo sensor shaft, in order to precisely measure its absolute position: any backlash in the system can then be compensated. Furthermore, for re-usability in other projects, the main shaft is hollow. It allows to mount a slip ring and consequently use the system in continuous turning mode. To increase compactness, it can be removed, because  $180^\circ$  rotation is sufficient to get a full 3D scan with the yaw scan top method. The sensor was designed to allow turning the sensor with the platform with any inclination regarding gravity.

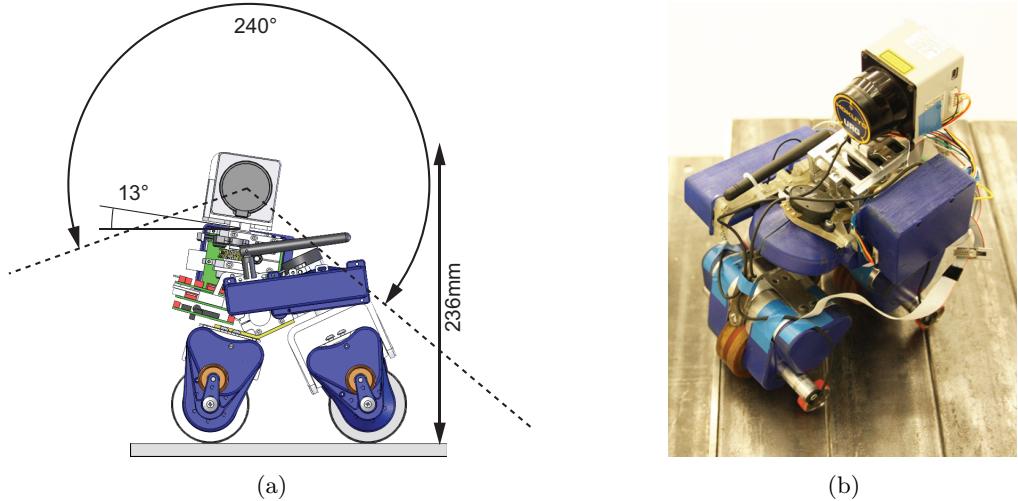


Figure 6: 3D range finder (rotating Hokuyo URG-04LX) mounted on the MagneBike robot.  
a) CAD model and field of view. b) The MagneBike equipped with the 3D LADAR.

The final assembly on the MagneBike is shown in Figure 6. The blind zone is located below the robot, where the scanner would scan the robot body. The  $13^\circ$  inclination allows to optimally scan the surroundings of the robot and favors areas in the driving direction. The total size of this version however slightly exceeds the specified 200mm (smallest diameter size). The main characteristics of the 3D LADAR are summarized in Table 1.

One can see that the sensor resolution satisfies the requirements presented in Section 2.1. With 3mm resolution at 0.5m, this resolution allows us to detect obstacles with a size of 10mm: there are at least three points on the target obstacle. A 3D scan taken in the real environment with full resolution -  $0.36^\circ$  resolution on elevation and azimuth angles - and with down-sampling to 2.5% of the data is shown in Figure 7. We can see the high density of points in areas close to the scanner (origin indicated by arrows) as well as the blind zone below the robot.

Table 1: 3D LADAR characteristics

Parameter	Value
Size of mechanics (without Hokuyo), W×L×H	50 × 67.5 × 55mm <sup>3</sup>
Mass with / without Hokuyo	0.34kg / 0.18kg
Backlash on the main shaft	<1°
Absolute encoder resolution	0.09°
Hokuyo URG-04LX resolution	0.36° (3mm at 0.5m)
3D scanner resolution	Adjustable
3D scan time (for 0.36° resolution)	50s (180° rotation)
Number of points (for 0.36° resolution)	683 × 500 = 341,500 points
Maximum rotational speed	3rpm

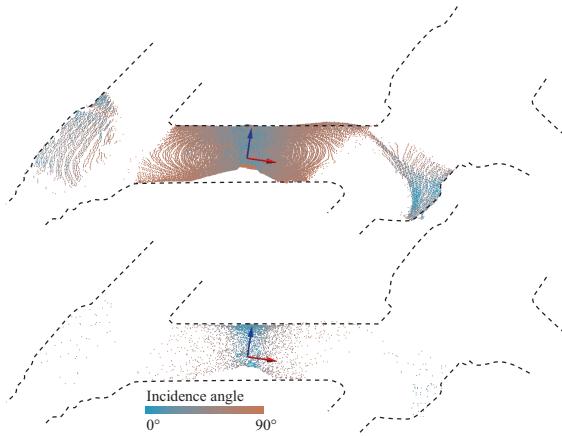


Figure 7: 3D scan taken in real environment (cut views) with full resolution (0.36°) (top) and with down-sampling to 2.5% of the points (bottom). The point color represents the laser incidence angle on the surface. The arrows indicate the origin of the 3D scan. To ease the understanding, an approximate sketch of the 2D contour is drawn. For details about the environment that was scanned, refer to Figure 18.

## 4 Localization: model and tools

The 3D range finder having been presented, this section describes the tools that are used to implement steps (a) and (c) of the localization strategy proposed in Section 2.3, respectively the 3D odometry and the local registration algorithm.

### 4.1 MagneBike description and model

As illustrated in Figures 8 and 9, the MagneBike consists of seven rigid bodies: the front and rear wheel units integrating lifter/stabilizer mechanisms, the steering unit, the free fork and the 3D Hokuyo LADAR. To fully define the state of the articulated and deformable system and its interaction with the environment through the two magnetic wheels, ten coordinate frames are defined (Figure 9):

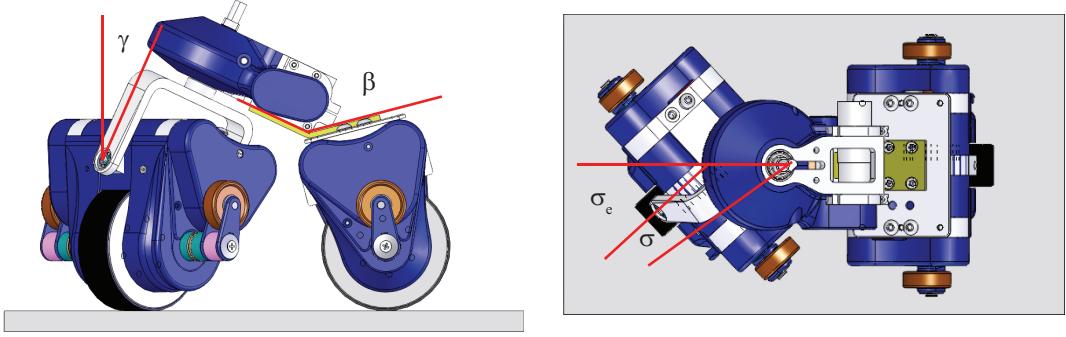


Figure 8: MagneBike description (without 3D LADAR): flexible joint between wheel units ( $\beta$ ), free joint on the fork ( $\gamma$ ) and steering angle ( $\sigma$  on the steering axis and  $\sigma_e$  on the plane).

- C and D: coordinate frames of the rear (resp. front) wheel contact point. They describe the interaction between the robot and its environment: x (forward) and y (side) axis are tangent to the surface while the z-axis is normal to it. The contact angles are defined as  $\delta_r$  and  $\delta_f$ .
- A and B: coordinate frames of the rear (resp. front) wheel units. They are centered on the wheel axis, with z-axis pointing vertically when the robot stands on a horizontal flat surface (dashed line and black dots in Figure 9). In that position  $\delta_r = 0$  and  $\delta_f = 0$ .
- S: steering unit coordinate frame. The steering unit is linked to the rear wheel unit through a flexible joint. It is used to measure internal forces by the very small robot deformation, in order to control the relative wheel speeds (Tâche et al., 2009). The flexible joint is described by the angle  $\beta$ . Both the accelerometer frame I and the 3D scanner frame H are fixed to the steering unit body.
- H: Hokuyo 3D scanner. The orientation of the scanner is given by coordinate frame H. It is tilted from angle  $\phi$  regarding frame S.
- F: fork coordinate frame. The fork, mounted on the steering axis, is rotated from an angle  $\sigma$  around the z-axis of frame S.  $\sigma$  is measured with an absolute encoder mounted on the steering shaft. Frame F is positioned on the top of the fork.
- J: free joint coordinate frame. The free fork angle  $\gamma$  (rotation around the x-axis) describes the free joint between the fork and the front wheel unit. There is no sensor mounted on this axis. This coordinate frame is actually on the same rigid body as B, but it is introduced for readability reasons.
- I: inclinometer (accelerometer) coordinate frame. It is fixed to the steering frame, but is necessary to define the accelerometer position and orientation on the robot.
- R: robot coordinate frame. This frame is used to integrate the robot position. It is positioned between frames C and D (robot on a flat surface) at a distance  $L_R = L_{w0}/2$  from C.

The coordinate frames of the front and rear lifters are not represented, because they are not important for localization purposes. All five locomotion DoF (wheels, lifters and steering) are equipped with relative optical encoders mounted on the actuators shafts.

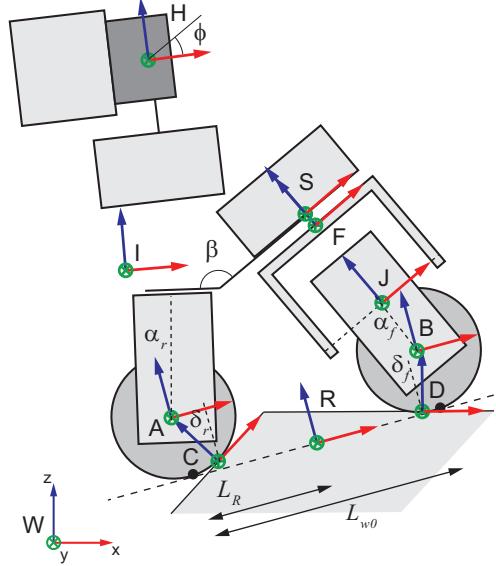


Figure 9: MagneBike model: coordinate frames definition. x, y and z axis are represented by red, green and blue arrows.

## 4.2 2D odometry

Because 3D odometry is an extension of 2D odometry, we start by describing the MagneBike 2D odometry model illustrated in Figure 10. It is simpler to understand and helps the reader to get an insight into the 3D model. Assuming that the robot does not slip, the kinematic model of the bicycle robot is given by its linear velocity  $\mathbf{v}_W$  in the world frame and its rotational velocity on the plane  $\omega$ :

$$\mathbf{v}_W = R_z(\Psi) \cdot \mathbf{v}_R = \begin{bmatrix} \cos \Psi & -\sin \Psi \\ \sin \Psi & \cos \Psi \end{bmatrix} \cdot \begin{bmatrix} v_R \cos \sigma_R \\ v_R \sin \sigma_R \end{bmatrix} \quad (1)$$

$$\omega = \frac{v_R}{R_{ICR,R}} = \frac{\sin \sigma_R}{L_R} v_R \quad \sigma_R = \arctan \left( \frac{L_R \cdot \tan \sigma_e}{L_w} \right) \quad (2)$$

with  $\mathbf{v}_R$  representing the robot velocity in the robot frame R,  $\sigma_e$  the effective steering angle of the wheel,  $R_{ICR,R}$  the distance of the robot frame to the instantaneous center of rotation (ICR) and  $\Psi$  the orientation of the robot in the world frame.

The position can be estimated by integrating the robot position and orientation in the world frame:

$$\mathbf{p}_{k+1} = \mathbf{p}_k + d\mathbf{p}_k = \mathbf{p}_k + R_z(\Psi_k) \cdot \begin{bmatrix} \cos \sigma_R \\ \sin \sigma_R \end{bmatrix} \cdot dS_R \quad (3)$$

$$\Psi_{k+1} = \Psi_k + d\omega_R = \Psi_k + \frac{\sin \sigma_R}{L_R} \cdot dS_R \quad (4)$$

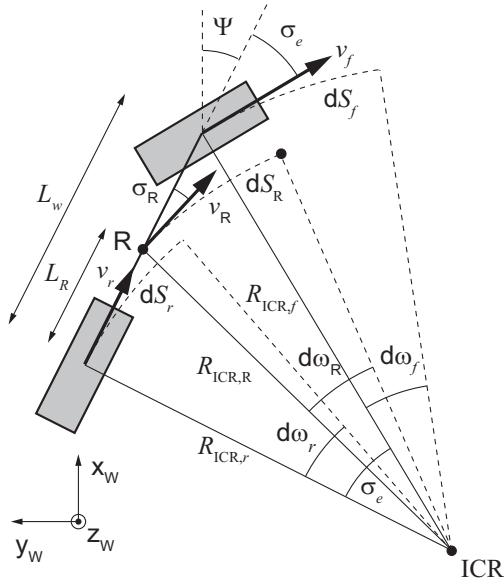


Figure 10: 2D odometry model for the MagneBike (top view).

$$dS_R = \frac{dS_f \cdot L_R \cdot \sin \sigma_e}{L_w \cdot \sin \sigma_R} \quad (5)$$

with  $dS_f$  representing the distance traveled by the front wheel,  $dS_R$  the corresponding distance traveled by the robot and  $L_w$  the wheel distance.

Because the steering axis is not perpendicular to the surface, the wheel distance  $L_w$  and effective front wheel steering angle  $\sigma_e$  are functions of the steering angle  $\sigma$  (measured on the steering axis) and the robot deformation  $\beta$ . Given that the robot state is not fully determined for each steering position  $\sigma$  (contact angles  $\delta_f$ ,  $\delta_r$  and the fork angle  $\gamma$  are unknown and not measurable),  $L_w$  and  $\sigma_e$  have to be computed numerically. The two following mechanical constraints are used:

- Constraint 1: because the surface is flat, the z-axis of coordinate frames C and D always remain parallel and perpendicular to the surface. The z-axis of D expressed in coordinate frame C or D is equal to  $\mathbf{e}_z$  (Equation 6). The rotation matrix from frame C to frame D is a pure rotation around  $z$  of angle  $\sigma_e$  (Equation 7):

$$\mathbf{z}_D|_C = R_{CD} \cdot \mathbf{z}_D|_D = R_{CS} \cdot R_{SF} \cdot R_{FJ} \cdot R_{JD} \cdot \mathbf{e}_z = \mathbf{e}_z \quad (6)$$

$$R_{CD} = R_y(-\delta_r + \alpha_r + \beta - \pi) \cdot R_z(\sigma) \cdot R_x(\gamma) \cdot R_y(\alpha_f + \delta_f) = R_z(\sigma_e) \quad (7)$$

- Constraint 2: on a flat surface, the contact point of the front wheel is at distance  $L_w$  on the x-axis of coordinate frame C, respectively the vector CD expressed in C is equal to  $L_w \cdot \mathbf{e}_x$ :

$$\mathbf{CD}|_C = \mathbf{CA}|_C + R_{CA} \cdot (\mathbf{AS}|_A + R_{AF} \cdot (\mathbf{SJ}|_F + R_{FJ} \cdot (\mathbf{JB}|_J + R_{JB} \cdot \mathbf{BD}|_B))) = L_w \cdot \mathbf{e}_x \quad (8)$$

Once the equations system (Equation 6 and 8) with the four unknowns ( $L_w$ ,  $\delta_f$ ,  $\delta_r$  and  $\gamma$ ) has been solved, the effective steering angle  $\sigma_e$  can be computed using rotation matrix  $R_{CD}$  (from Equation 7):

$$\sigma_e = \arccos(R_{CD[1,1]}) = \arcsin(R_{CD[2,3]}) \quad (9)$$

The values of the effective steering angle  $\sigma_e$ , the robot length  $L_w$ , the fork inclination  $\gamma$  and the contact angles  $\delta_f$  and  $\delta_r$  in function of the steering angle  $\sigma$  (for  $\beta = \beta_0 = 140^\circ$ ) are then precomputed and stored in a lookup table. They allow to integrate the robot pose using Equations 2 to 5.

### 4.3 3D odometry for the MagneBike

If the robot is moving around on an uneven surface, it would be necessary to measure the fork inclination  $\gamma$  and the contact angles  $\delta_f$  and  $\delta_r$  to get an exact model for 3D odometry (conditions 1 and 2 from Equations 6 and 8 are only valid on flat surfaces). Integrating more sensors to measure these values would make the system very complex to improve odometry that is in any case prone to integration errors. Instead we present a simplified 3D odometry model that is used to track the robot pose between 3D scans and to provide an initial guess for the 3D scan matching. The model makes the following assumptions:

- The robot moves on a flat plane  $\Pi_k$  of orientation  $R_k$  (rotation matrix from plane to world) during the short integration step  $dt$  between the discrete sampling time  $t_k$  and  $t_{k+1}$ . The 2D odometry model is used to compute the robot displacement  $d\mathbf{p}_k$  on plane  $\Pi_k$  (Figure 11). This assumption allows to estimate the robot state, respectively the contact angles, the fork angle as well as the robot length as presented in the 2D model. Knowing that the environment is comprised of curved surfaces, this assumption is valid in most cases, *e.g.* when the robot is driving longitudinally in pipe structures or surface with high curvatures. The model is however not accurate when the robot negotiates step obstacles, because of the poor approximation of the contact angles.
- The robot orientation is updated using the robot rotation on the plane (due to steering) and the readings from the 3D accelerometer that is assumed to only measure the gravitational acceleration. This assumption is applicable, because the robot moves slowly. Since the accelerometer only allows for the observation of two DoF (plane inclination regarding gravity without information about the robot orientation around the gravity vector  $\mathbf{g}$ ), the robot orientation estimation model is augmented with a filter estimating the local surface curvature.
- The robot body is assumed to be rigid ( $\beta = \beta_0 = 140^\circ$ ), even if there is a bit of flexibility between the wheel units. This assumption is valid, because the wheel speed controller tries to cancel this deformation as described by (Tâche et al., 2009).
- The front wheel displacement  $dS_f$  is used to compute the robot displacement  $dS_R$  (Equation 5) and the rotation  $\omega$  (Equation 2). The rear wheel displacement  $dS_r$  is actually prone to a singularity: when the steering angle is close to  $90^\circ$  (*e.g.* robot doing a turn on spot movement), the rear wheel gets close to the ICR and its displacement tends to zero.

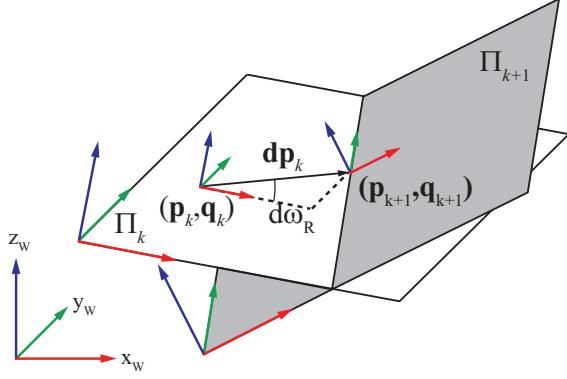


Figure 11: 3D odometry model: during integration step  $t_k$  to  $t_{k+1}$ , the robot moves by  $\mathbf{dp}_k$  and rotates from  $d\omega_R$  on the plane  $\Pi_k$ .

#### 4.3.1 Position and orientation estimation

For the position update, the small displacement  $\mathbf{dp}_k$  on the plane  $\Pi_k$  is computed with the 2D odometry model extended to 3D with  $\mathbf{dp}_k = (\cos \sigma_R, \sin \sigma_R, 0)$ . Regarding the orientation, the available measurements are the robot rotation on the plane due to steering ( $\omega_R$ ) between time  $t_k$  and  $t_{k+1}$  and an acceleration measurement  $\mathbf{a}_R$  taken at time  $t_{k+1}$ . The corresponding rotations parameterized as axis angles are respectively the robot rotation expressed in the robot frame R:

$$d\boldsymbol{\omega} = d\omega_R \cdot \mathbf{e}_z \quad (10)$$

and the plane rotation measured in the world frame W that can be approximated as:

$$d\boldsymbol{\omega}_a = \left( R_k \cdot \frac{\mathbf{a}_R}{\|\mathbf{g}\|} \right) \times \frac{\mathbf{g}}{\|\mathbf{g}\|} \quad (11)$$

with  $R_k$  the robot/plane orientation at time  $t_k$  and  $\mathbf{a}_R$  the acceleration measured at time  $t_{k+1}$ . This approximation is valid, because the robot moves (resp. rotates) slowly and the state is updated at a high rate (20Hz). This first assumption implies that only small rotations are measured, the cross product is then a correct approximation of these rotations. Furthermore, it can also be considered that the accelerometer only measures the gravitational acceleration. High accelerations due to impacts on obstacles are easily filtered.

The robot orientation can then be updated by adding these two contributions:

$$R_{k+1} = R(d\boldsymbol{\omega}) \cdot R_k \cdot R(d\boldsymbol{\omega}_a) = e^{[d\boldsymbol{\omega}]_x} \cdot R_k \cdot e^{[d\boldsymbol{\omega}_a]_x} \quad (12)$$

$R(d\boldsymbol{\omega})$  denotes the rotation matrix equivalent to the small rotation  $d\boldsymbol{\omega}$ , while the symbol  $[.]_x$  is used to denote the cross product matrix (a skew-symmetric matrix). Equation 12 can be written in quaternion form (quaternions are preferred for the implementation):

$$\mathbf{q}_{k+1} = \mathbf{q}(d\boldsymbol{\omega}_a) \oplus \mathbf{q}_k \oplus \mathbf{q}(d\boldsymbol{\omega}) \quad (13)$$

with  $\mathbf{q}(d\boldsymbol{\omega})$  the quaternion representing the small rotation  $d\boldsymbol{\omega}$  and  $\oplus$  the quaternion multiplication operator.

The limitation of this approach is that the 3D accelerometer only provides partial information about the robot orientation, respectively the plane inclination regarding gravity, but not the

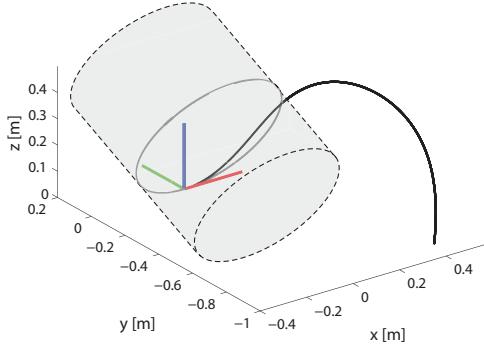


Figure 12: 3D odometry model limitation. Simulation of a circumferential path in an inclined pipe. Pipe diameter = 0.6m, inclination =  $30^\circ$ , start position =  $(0,0,0)$ . The ground truth is plotted in grey and 3D odometry in black.

robot orientation around the gravity vector. The robot heading is only integrated when the rotational velocity of the robot on the plane is not null. This effect can be illustrated by two extremum cases. If the robot follows a circular trajectory on a flat plane, the heading is fully defined by integrating  $\omega$ . If the robot follows a circumferential trajectory in an inclined tube (it does not need to steer,  $\omega = 0$ ), the acceleration information is insufficient to determine the robot orientation. The model does not provide any heading component which actually remains constant. This limitation leads to the trajectory depicted in Figure 12 which was generated from simulation data (exact encoder and accelerometer measurements without noise).

#### 4.3.2 Orientation estimation: extended model

Knowing that the environment is mainly built out of cylindrical and spherical sections, the model is enhanced with a correction term from an estimate of the local surface curvature. Since the cylinders can have any inclination regarding gravity, the curvature  $\kappa$  is estimated in the robot frame, while the corresponding rotation error  $\Delta r$  is estimated in the world frame (measurement frame). Because the curvature changes over distance, its prior model is based on an exponentially decaying function of the traveled distance (Equation 15). Thus the curvature is effectively estimated over a limited distance window of past data with the distance constant  $\tau_S$ . This model also maintains the fact that the curvature remains unchanged when the robot does not move ( $dS_R = 0$ ).

$$\Delta \mathbf{r} = \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_z \end{bmatrix}_W \cdot dS_R = R \cdot \boldsymbol{\kappa}_R \cdot dS_R \quad (14)$$

$$\dot{\boldsymbol{\kappa}}_R(t) = -\frac{1}{\tau_S} \frac{dS_R}{dt} \boldsymbol{\kappa}_R(t) \quad \boldsymbol{\kappa}_R(t) = e^{-\frac{1}{\tau_S} dS_R} \quad (15)$$

The problem then consists in estimating the robot orientation ( $R_k$  or  $\mathbf{q}_k$ ), *i.e.* the following continuous time error state model:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \Delta\dot{\mathbf{r}} \\ \dot{\boldsymbol{\kappa}}_R \end{bmatrix} = \begin{bmatrix} 0 & R \\ 0 & -\frac{1}{\tau_S} \cdot I \end{bmatrix} \cdot \begin{bmatrix} \mathbf{r} \\ \boldsymbol{\kappa}_R \end{bmatrix} \cdot \frac{dS_R}{dt} + \tilde{\mathbf{v}} \left( \frac{dS_R}{dt} \right) = A(t) \cdot \mathbf{x}(t) + \tilde{\mathbf{v}} \left( \frac{dS_R}{dt} \right) \quad (16)$$

or its discrete version:

$$\mathbf{x}_{k+1} = \begin{bmatrix} I & R \cdot dS_R \\ 0 & \left(1 - \frac{dS_R}{\tau_S}\right) \cdot I \end{bmatrix} \cdot \mathbf{x}_k + \mathbf{v}_k(dS_R) = F_k \cdot \mathbf{x}_k + \mathbf{v}_k(dS_R) \quad (17)$$

The process noise is modeled as a function of the robot displacement and time independent, because  $\mathbf{x}$  is steady when the robot stands still. The error state  $\mathbf{x}$  is estimated with an extended Kalman filter (EKF) using the following residual measurement function:

$$\mathbf{h}(\mathbf{a}_R, \mathbf{x}) = \left( R \cdot \frac{\mathbf{a}_R}{\|\mathbf{g}\|} \right) \times \frac{\mathbf{g}}{\|\mathbf{g}\|} \quad (18)$$

This function effectively measures the rotation error based on the current acceleration measurement  $\mathbf{a}_R$  and the robot orientation estimation  $R$ . The residual function is null if the orientation prediction is correct and if the acceleration measurement is exact. Hence, the residual measurement is used to update the error state  $\mathbf{x}$ , namely the correction term that is applied to update the robot orientation.

#### 4.3.3 Sensor model and covariance matrices of the EKF

The error state and measurement models having been defined, we define the covariance matrix of the measurement noise  $\tilde{R}$ , the observation model  $H_x$  and the process noise covariance matrix  $Q$ . With these definitions, the specification of the extended Kalman filter can be completed, namely the state covariance  $P_{k+1|k} = F_k \cdot P_{k|k} \cdot F_k^T + Q_k$  and the innovation covariance matrix  $S_{k+1} = H_{xk+1} \cdot P_{k+1|k} \cdot H_{x,k+1}^T + \tilde{R}_{k+1}$ .

The accelerometer is assumed to be rotationally symmetric without any cross-correlation between the three acceleration components. The sensor covariance matrix  $\Sigma_{\mathbf{a}_R}$  and its implication  $\tilde{R}$  on the residual measurement model can be written as:

$$\tilde{R} = H_{\mathbf{a}_R} \cdot \Sigma_{\mathbf{a}_R} \cdot H_{\mathbf{a}_R}^T \quad \Sigma_{\mathbf{a}_R} = I \cdot \sigma_{\mathbf{a}_R}^2 \quad (19)$$

with:

$$H_{\mathbf{a}_R} = \frac{\delta \mathbf{h}}{\delta \mathbf{a}_R} = -\frac{[\mathbf{g}]_\times}{\|\mathbf{g}\|} \cdot R \quad (20)$$

The observation model  $H_x$  necessary to compute  $S$  is given by:

$$H_x = \frac{\delta \mathbf{h}}{\delta \mathbf{x}} = \begin{bmatrix} \frac{\delta \mathbf{h}}{\delta \Delta \mathbf{r}} & \frac{\delta \mathbf{h}}{\delta \boldsymbol{\kappa}_R} \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \quad (21)$$

The result can be derived by approximating the residual function as  $\mathbf{h} = \left( \mathbf{d}\mathbf{r} \cdot R \cdot \frac{\mathbf{a}_R}{\|\mathbf{g}\|} \right) \times \frac{\mathbf{g}}{\|\mathbf{g}\|}$  with  $\mathbf{d}\mathbf{r} = e^{[\Delta\mathbf{r}]_x} \approx I + [\Delta\mathbf{r}]_x$ .

The process noise  $\mathbf{v}$  is assumed to be zero-mean Gaussian. In order to avoid complexity we also assume no cross-correlation as well as rotational symmetry in the rotation error and curvature model. Because the robot state is fully defined when the robot stands still, the process noise is set proportional to the distance traveled  $dS_R$ . The state covariance matrix  $P$  then only increases when the robot is moving. The covariance of the process noise is then defined as:

$$Q = E \{ \mathbf{v}\mathbf{v}^T \} = \begin{bmatrix} I \cdot \sigma_r^2 & 0 \\ 0 & I \cdot \sigma_\kappa^2 \end{bmatrix} \cdot dS_R \quad (22)$$

The variance  $\sigma_r^2$  can be interpreted as the noise level on the orientation estimation due to surface roughness. If the surface has a constant inclination, the curvature  $\kappa$  is null and the only rotation contribution is noise. The variance  $\sigma_\kappa^2$  represents the variation of the environment curvature and is assumed as normal with zero mean. This is approximately true, since the robot spends most of its time driving along the longitudinal axis of pipe-type structures (curvature close to 0). The likelihood of driving on curved surfaces, typically in intersections or on circumferential paths, is lower. Furthermore the probability to overcome step obstacles, equivalent to driving on surfaces with very high curvatures, is the lowest.

Because the rotation about  $z$  ( $\Delta r_z$ ) is in the null space of  $\mathbf{h}(\mathbf{a}_R, \mathbf{x})$ , it is unobservable and the corresponding covariance continuously grows with distance. Furthermore, since the surface curvature has only two components (along  $x$  and  $y$ ), the error state model is simplified to its four dimensional version  $\mathbf{x} = (\Delta\mathbf{r}, \boldsymbol{\kappa}) = (\Delta r_x, \Delta r_y, \kappa_x, \kappa_y)$ . The matrices defined above then become:

$$\begin{aligned} \tilde{R} &= I_{2\times 2} \cdot \sigma_{\mathbf{a}_R}^2 & H_{\mathbf{x}} &= [ I_{2\times 2} \ 0 ] \\ Q &= \begin{bmatrix} I_{2\times 2} \cdot \sigma_r^2 & 0 \\ 0 & I_{2\times 2} \cdot \sigma_\kappa^2 \end{bmatrix} \cdot dS_R \end{aligned} \quad (23)$$

All matrices having been defined, the robot orientation estimation procedure consists of first predicting the robot orientation at time  $t_k$  using the error state estimate  $\hat{\mathbf{x}}_{k|k}$ :

$$\hat{\mathbf{q}}_{k+1|k} = \mathbf{q}(\Delta\mathbf{r}_{k|k}) \oplus \hat{\mathbf{q}}_{k|k} \oplus \mathbf{q}(\boldsymbol{\kappa}_{k|k} \cdot dS_R) \quad (24)$$

with

$$\Delta\mathbf{r}_{k|k} = \begin{bmatrix} \Delta\hat{\mathbf{r}} \\ 0 \end{bmatrix} \quad \boldsymbol{\kappa}_{k|k} = \begin{bmatrix} \hat{\boldsymbol{\kappa}} \\ \kappa_z \end{bmatrix} \quad (25)$$

The robot's rotation around the  $z$ -axis of the robot frame due to steering  $d\omega = \frac{\sin \sigma_R}{L_R} \cdot dS_R$  (Section 4.2) can be interpreted as the  $z$  component of the curvature vector, namely the path curvature with  $\kappa_z = \frac{\sin \sigma_R}{L_R}$ . The error state is then updated with the Kalman filter equations, using the residual measurement function (the first two dimensions of Equation 18) and dynamically updating the innovation  $S$  and error state  $P$  covariance matrices.

## 4.4 Local maps registration: pairwise scan matching

If two 3D scans taken at different locations have some overlap, it is possible to find a rigid body transformation between these two scans. A common way to solve this problem is to use the ICP (Iterative Closest Points or Iterative Corresponding Points) proposed by (Besl and McKay, 1992). In the literature, many different variants of the algorithm are proposed and compared (Rusinkiewicz and Levoy, 2001). We then give a brief overview of the five ICP algorithm steps and the variant which was used.

1. *Selection*: this optional first step consists in selecting some points in one or both input scans  $A$  and  $B$ . The main goal is to reduce the computation time of step 2 which is the most demanding. Examples are uniform sub-sampling, random sampling or selection of points that contain much information (high intensity gradient, color). Our implementation uses random sampling because of its low computational cost. The selection and validation of an optimal percentage ratio is discussed in more detail in Section 5.3.1.
2. *Correspondence finding*: for each point in scan  $A$ , the goal is to find the closest (resp. corresponding) point in scan  $B$ . This step is the most expensive regarding computation time. Our implementation uses a kd-tree (Bentley, 1975) which reduce the complexity to  $O(N \log M)$  with  $N$  and  $M$  the number of points in scan  $A$  and  $B$ .
3. *Correspondence pair weighting*: this optional step consists in assigning some weights to the corresponding point pairs found in step 2. The weights can for instance depend on the point-to-point distance or the compatibility of the normal vectors. Our implementation does not use any weighting algorithm.
4. *Correspondence rejection*: this additional optional step consists in rejecting some corresponding point pairs (somehow assigning weight 0 to bad correspondences), in order to increase the accuracy of the 3D transformation. This mainly helps to remove erroneous pairing due to partial overlap of scans, sensor noise and dynamic obstacles. This can be done by rejecting pairs with a point-to-point distance above a threshold, by rejecting a certain percentage of pairs based on a metric criterion or by rejecting pairs which are not consistent with neighboring pairs. Our implementation uses a Relative Motion Threshold (Pomerleau et al., 2009) which adapts the rejection threshold depending on the convergence velocity of ICP.
5. *Error metric and minimization*: once two sets of correspondence points have been found and sorted, the goal is to find a rigid-body transformation  $T$  that minimizes an error metric. This metric is often the mean squared distance between points in scan  $A$  and its corresponding transformed points in scan  $B$ , it is known as the point-to-point metric. Faster convergence can be achieved using point-to-plane metric (Rusinkiewicz and Levoy, 2001) which is used in our implementation. The later metric computes the mean squared distance from points in scan  $A$  and the plane containing the corresponding points in  $B$  and oriented perpendicular to its normal vector. The normal vector is computed in a preprocessed step using the 20 nearest neighbors of each point in scan  $A$ . In the case of point-to-plane error, no close form solution exists for 3D point clouds. The selected method linearizes the problem, assuming small angle variations in between iterations, and solves the equation system using a

Cholesky factorization.

Steps 2 to 5 are iteratively repeated until a stopping criterion is reached, which often consists in checking that the difference between the mean squared distance used in step 5 of the current iteration and the previous one falls below a fixed threshold or that a maximum number of iterations has been reached. For faster convergence and increasing the probability to converge to the global minimum - convergence to a local minimum is ensured and proven by (Besl and McKay, 1992) - the output of 3D odometry can be used as an initial guess or prior for the first iteration.

#### 4.5 Combining 3D odometry with 3D scan registration

As mentioned previously, the 3D scans and their registration is not only used to construct a global 3D model of the environment, but also allows us to refine the position of the robot. 3D odometry and scan matching are combined as follows:

- 3D odometry is used to continuously track the robot position between consecutive scans. The transformation given by odometry ( $T_{\text{ODO},ij}$  in Figure 13) is used as a prior for ICP.
- Since relative transformations given by scan matching ( $T_{\text{ICP},ij}$ ) are more precise than 3D odometry which drifts faster, the robot trajectory computed with 3D odometry is updated with the transformation returned by scan matching (thick solid trajectory in Figure 13).

Figure 13 shows a schematic where all transformations are expressed in the same frame (for instance in the odometry frame R). The scan registration transformations are computed however in the 3D range finder frame H. This is rather important, because both frames do not move of the same quantity when the surface is uneven. More details about this are given in Section 5.6.2 that discusses the results.

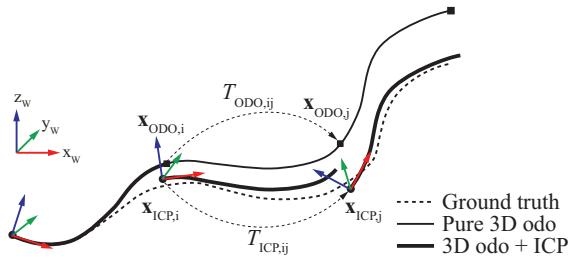


Figure 13: 3D odometry and scan matching combination (schematic model). The ground truth is represented by the dashed trajectory. The trajectory computed with pure 3D odometry is drawn with a thin solid line, while the thick solid line represents the trajectory combining scan matching with odometry. The dots show the scan positions computed by scan matching, while the squares show their positions if only odometry were used.

Even if scan matching provides precise results, it only gives relative transformations between scan positions and alignment errors are still accumulated along the trajectory. The relative

orientation to the world frame is thus slowly lost, if only the start orientation is known or taken into account. This is critical for the user, since it is not only important to know the orientation of the robot relative to the environment, but also the orientation of the environment relative to the world frame. The orientation of the 3D visualization is correspondingly adjusted using the accelerometer readings which provide reliable and absolute measurements with respect to the gravity vector. The map may then be distorted, but locally aligned with the gravity frame. Ideally all measurements should be used and optimized, but this is not a trivial problem. The complete fusion actually implies to develop detailed error propagation models for both scan matching and 3D odometry. This complex modeling is planned as future work.

## 5 Experimental results

This section aims at evaluating the performance of the localization strategy and tools proposed in this paper - *i.e.* 3D odometry, 3D scan registration as well as their combination - through experiments with the MagneBike robot. 3D odometry is first tested on laboratory reference environments such as a pipe of constant diameter or step-type obstacles (Section 5.1). These well defined geometrical structures allow us to easily repeat experiments for which the ground truth trajectory is available and to evaluate in detail the performance of the model. Section 5.2 then describes the field experiments that were performed to characterize the localization procedure in real conditions. Because in this case the exact ground truth is not available, the result of scan registration is used as limited ground truth for the evaluation of 3D odometry. The characterization of 3D scan matching is then first presented in Section 5.3 followed by 3D odometry characterization in Section 5.4. The combination of 3D odometry with 3D scan registration, actually the final result of the localization and mapping procedure, is presented in Section 5.5. Design guidelines and further results are finally discussed in Section 5.6.

### 5.1 3D odometry: implementation, model validation and evaluation

Because the 3D odometry model is the most significant contribution and a low level component of the proposed localization procedure, it is first characterized in detail through experiments in known laboratory environments for which ground truth is available.

#### 5.1.1 Filter parameter setting

Regarding 3D odometry, the first step is to set the parameters of the Kalman filter used to estimate the robot orientation. These parameters are determined by characterizing the different sources of noise on sensors and on the system.

#### Acceleration sensor model

Because the accelerometer (frame I) is not mounted parallel to the odometry frame R, its orientation relative to the plane ( $\delta_r$ ) is changing with the steering angle ( $\sigma$ ). The acceleration

measured in the inclinometer frame has to be transformed into the plane (respectively robot) frame R with:

$$\mathbf{a}_R = R_{RR0} \cdot R_{R0I} \cdot \mathbf{a}_I \quad (26)$$

$R_{R0I}$  is the calibration matrix that corrects the offset between the inclinometer and a horizontal surface when the steering angle is null. Assuming that there is no heading offset,  $R_{R0I}$  can be measured through accelerometer measurements when the robot is placed on a horizontal surface.  $R_{RR0}$  corrects the offset due to the steering, respectively a rotation of  $\delta_{r0} - \delta_r$  around the y-axis.

The noise on the acceleration vector was measured in some accelerometer calibration data sets in which the sensor was set in many different orientations. It was also measured with the sensor mounted on the robot to confirm that the robot's electronic does not generate extra noise on the measurements. The noise variance  $\sigma_{\mathbf{a}_R}^2$  on the acceleration vector  $\left(\frac{\mathbf{a}_R}{\|\mathbf{g}\|}\right)$  was measured equal to  $0.0063^2$  [rad $^2$ ].

### Setting the process noise matrix Q

As mentioned in Section 4.3.3,  $\sigma_r^2$  can be interpreted as the noise level on the orientation estimation due to the surface roughness. To estimate it, the robot was driven longitudinally in a straight pipe and the noise on the acceleration vector was measured. Since the measured variance  $\sigma_m^2$  is the combination of the sensor and roughness noise, the pure sensor noise  $\sigma_{\mathbf{a}_R}^2$  has to be subtracted:

$$\sigma_r^2 = (\sigma_m^2 - \sigma_{\mathbf{a}_R}^2) / dS_R \quad (27)$$

The surface roughness noise variance  $\sigma_r^2$  was measured equal to  $0.035$  [rad $^2$ m $^{-1}$ ].

The variance  $\sigma_k^2$  representing the variation of the environment curvature is more difficult to estimate. As discussed in Section 4.3.3, the distribution is assumed Gaussian with zero mean. The typical range of radii of the cylinders goes from 200 to 700mm, respectively curvatures between 2 and 1.4 [rad m $^{-1}$ ]. When the robot overcomes step obstacles (typically 5 to 10cm high), this is equivalent to drive on surfaces with high curvatures, *i.e.* curvatures up to  $+/-10$  [rad m $^{-1}$ ]. This result is obtained by simulating the robot trajectory on a step obstacle and looking for the highest curvature on the path. If we consider that this maximum value corresponds to the  $3\sigma_k$  value of the normal curvature distribution, its variance  $\sigma_k^2$  can be estimated as  $11$  [rad $^2$ m $^{-2}$ ].

### Filter parameters

The last parameter of the filter is the distance constant  $\tau_S$  of the curvature model (Equation 15). This parameter was first intuitively set to 0.2m (slightly bigger than a robot length). It was however empirically found that values around 0.01m provide a more stable estimation of the curvature. The final set of parameters used in the results presented hereafter is then:  $\sigma_{\mathbf{a}_R}^2=0.0063^2$  [rad $^2$ ],  $\sigma_r^2=0.035$  [rad $^2$ m $^{-1}$ ],  $\sigma_k^2=11$  [rad $^2$ m $^{-2}$ ] and  $\tau_S=0.01$  [m].

### 5.1.2 Model validation and evaluation

The 3D odometry model was first validated with simulation data. The simulation presented in Figure 12 is repeated and the robot trajectory perfectly fits with the ground truth. The same trajectory was also carried out with the MagneBike, *i.e.* a circumferential trajectory in a 0.6m tube inclined by 30°. The resulting 3D odometry trajectories without and with curvature estimation are plotted in Figure 14 with the ground truth in grey. It clearly shows the benefit of the model including the curvature estimation, but also the drift propagated over the 1.88m long path.

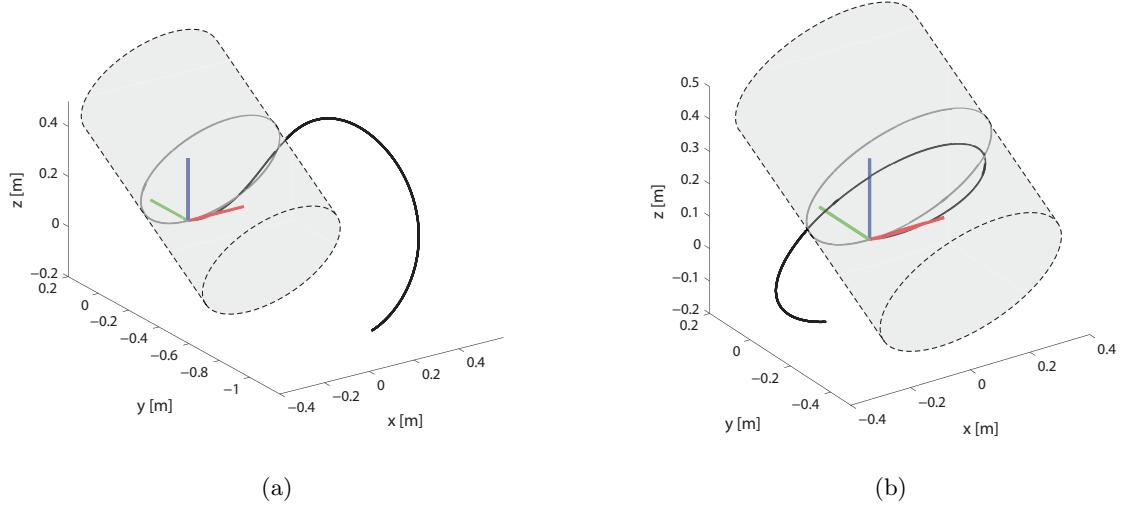


Figure 14: Orientation estimation model validation. 3D odometry (in black) on a circumferential path without (a) and with (b) curvature estimation. The pipe has a diameter of 0.6m and is inclined by 30°, the ground truth trajectory is plotted in grey.

For a more detailed characterization of 3D odometry, three types of trajectories were actually tested. The robot has first been tested on circumferential trajectories in a pipe with an inner diameter of 0.6mm. Experiments were done for different inclinations of the pipe, *i.e.* 0° (horizontal), 30° and 60°. The robot was also driven on helical trajectories (1m in length) in the same pipe with the same three inclinations. These first six experiments allow to show the performance of the 3D odometry model in a typical curved environment and full 3D trajectories with typically overhanging positions. The robot was finally climbing over a metallic box (80mm high, 417mm long) placed on a horizontal and a 30° inclined plane. These two last experiments allow to show the errors induced by the model approximations when the robot drives on discontinuous surfaces. For all the aforementioned tests, the ground truth trajectory was drawn on the environment and the robot was remote controlled to follow the drawn path. Each experience was repeated ten times, in order to get an error distribution model of the accumulated odometry error.

Figure 15 for instance shows five circumferential trajectories in a horizontal pipe (in black), while the ground truth trajectory is shown in grey. One can first see that all trajectories follow the shape of the ground truth pipe. This figure also provides an insight into the error propagation along the trajectory - the pipe curvature is first overestimated, then underesti-

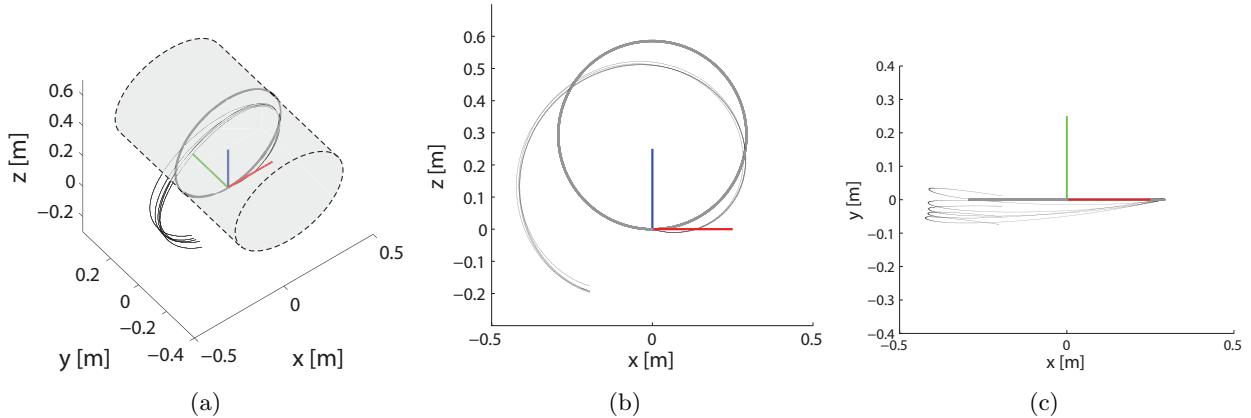


Figure 15: 3D odometry of five circumferential trajectories (in black) and ground truth trajectory plotted in grey. a) 3D view, b) side view, c) top view.

mated - and how trajectories differ. For a detailed quantitative evaluation of the accumulated errors, refer to Table 2 which provides an overview of the translation and rotation errors at the end of the different types of trajectories. Table 2 actually shows:

- $dx, dy, dz$ : the average of the absolute translation errors at the end of the 3D trajectory. The average is an appropriate descriptor of the translation error, because 3D errors are not randomly distributed. The errors are systematic (see Figure 15 for an example) and can be described by the centroid of all error measurements. This type of error distribution was observed for all data sets.
- $dt$ : the average of the absolute 3D translation errors (Euclidean distance).
- $d\theta$ : the average of the absolute rotation errors.
- $dS$ : the average of path lengths. The path length of each experiment is estimated by integrating the wheel encoders data (average of both wheels).

The standard deviation ( $\sigma$ ) is also given for  $dx$ ,  $dy$ ,  $dz$  and  $d\theta$ . Table 2 finally shows the translation and orientation error propagation rates, *i.e.*  $dt/dS$  and  $d\theta/dS$ .

From the table one can conclude that the translation error is between 0.05 to 0.14m per meter traveled, while the rotation error is between 2 to 11° per meter traveled. These are very good results considering that the system follows full 3D trajectories and that the localization problem is solved by using only two wheel encoders and a three axis accelerometer. 3D odometry then seems to be a good candidate to track the robot between scans and to be used as prior guess for the scan registration problem: these points are discussed in more detail in the following sections.

In Table 2 there is however an outlier, respectively the helical trajectory inclined by 60° for which the error is significantly higher. This case illustrates the main limitation of the odometry model that happens on sideways drives, respectively when the y-axis of the accelerometer is aligned with the gravity vector. In this particular case no orientation change can be detected by the accelerometer and the robot is somehow blind. This situation actually happens

Table 2: 3D odometry evaluation. Average and standard deviation of errors for eight different trajectories. Each experiment is repeated ten times.

<b>Trajectory, inclination</b>	$dx \pm \sigma_x$ [mm]	$dy \pm \sigma_y$ [mm]	$dz \pm \sigma_z$ [mm]	$dt$ [mm]	$d\theta \pm \sigma_\theta$ [°]	$dS$ [m]	$dt/dS$ [%]	$d\theta/dS$ [°/m]
Circle at 0°	-196±004	-31±036	-188±007	275	7±04	1917	14.4	3.5
Circle at 30°	-126±088	-40±021	-115±104	187	7±03	1876	10.0	3.6
Circle at 60°	-152±035	9±042	-3±003	157	20±09	1879	8.4	10.9
Helix at 0°	182±140	-52±097	-91±031	241	15±14	2170	11.1	7.1
Helix at 30°	-105±133	147±067	-69±015	220	22±13	2157	10.2	10.2
Helix at 60°	-623±135	437±186	-24±020	777	35±14	2149	36.1	16.1
Step at 0°	53±010	3±019	-25±016	63	3±01	1171	5.4	2.2
Step at 30°	88±009	23±029	-15±018	97	5±02	1186	8.1	4.0

at the beginning of the helical trajectory. The odometry estimation then quickly diverges from the real trajectory. The absolute worst case is a pure sideways trajectory such as a horizontal circumferential path illustrated in Figure 16. Without prior knowledge about the surface curvature (initialized to 0), the estimated trajectory is close to a horizontal straight line. The robot then cannot track its position between 3D scans using odometry, but 3D scanning and matching can still provide information to the system to locate the robot at scan positions. The distance between consecutive scans should therefore be smaller in such situations.

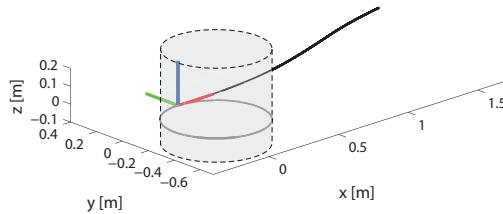


Figure 16: Limitation due to the 3D odometry model. Circumferential path in a vertical pipe with 0.6m of diameter. Pure 3D odometry in black versus ground truth in grey. The 3D scanner and ICP enables to solve that issue.

Figure 17 finally shows the step obstacle passing experiment with 30° inclination. It especially illustrates the influence of two model assumptions: 1) the use of the 2D odometry model (flat surface) to compute the contact angles: this model is not accurate on such obstacles where surfaces are perpendicular. 2) the continuity of the the surface curvature (filter

memory effect) that does not allow to model discontinuous surfaces such as edges. These assumptions lead to the smoothed trajectory shown in black on Figure 17. The grey trajectory shows the ground truth trajectory of the odometry frame ( $R$ ). The trajectory shape is not exact, but the final localization error is low as shown in Table 2 (8% in translation and  $4^\circ/\text{m}$  in rotation).

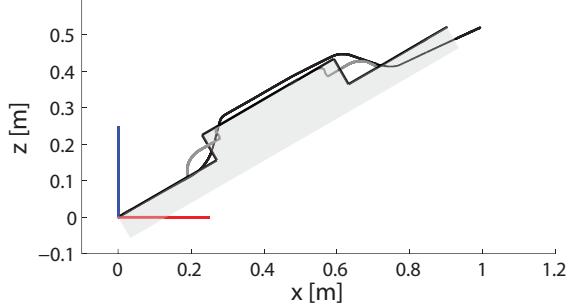
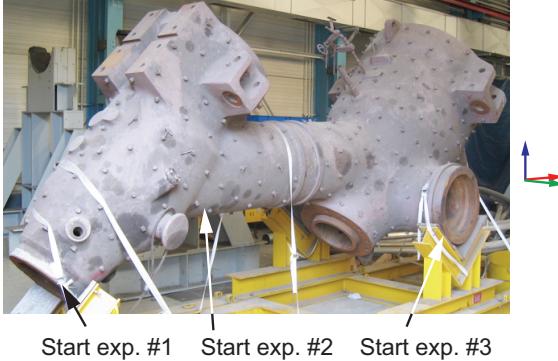


Figure 17: 3D odometry (in black) on a box-type obstacle. The figure also shows the ground surface as well as the ground truth trajectory of the odometry frame ( $R$ ) in grey.

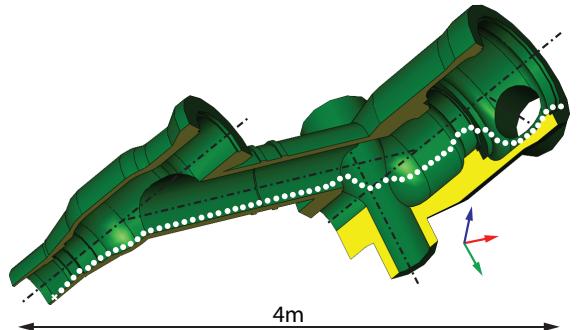
## 5.2 3D localization: experiments in real environment

In order to evaluate the performance of the localization concept in field conditions, the system was tested in a steam chest environment (extended version of the CAD model presented in Figure 3, right). The steam chest that is available for testing is shown in Figure 18(a). This part is actually out of power plant for repairing purposes, and hence has the advantage of being easily accessible. Furthermore, thanks to the multiple openings the robot can be tracked by the operator and easily retrieved in case of failure. It should be noted that at the current stage of the project (*i.e.* developing localization techniques for characterization and improvements) this is the only real application part available. Accessing other real facilities during outage time to get more data sets will only be allowed for a reliable industrialized version of the MagneBike.

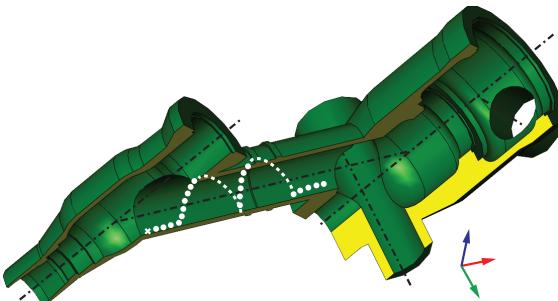
Figure 18 also shows CAD models of this environment and the three trajectories that were carried out (dotted paths). They were selected to be as various as possible and to cover a wide range of complexity regarding 3D localization. It should be mentioned that the CAD drawings are actually approximate models used to facilitate the understanding of the experiments presented hereafter, but they cannot be used as ground truth. In the first experiment [Figure 18(b)], the robot is mainly following longitudinal paths in the cylindrical sections. On this path the robot is mainly driving straight, nevertheless it has to avoid the big bottom hole and encounters several step obstacles. In the second experiment [Figure 18(c)], the robot follows a helical trajectory with two loops in the central cylinder. The robot then reaches a broad spectrum of 3D orientations, in particular upside down positions. The third experiment [Figure 18(d)] is the most complex trajectory. It has almost no straight sections, the robot has to avoid the bottom hole and then drives on curved and side inclined surfaces. The trajectory is also fully 3D and contains a circumferential path. The most interesting property of this trajectory is that it is a closed loop. The robot drives back close to its start position and scans the same pipe section from different points of views.



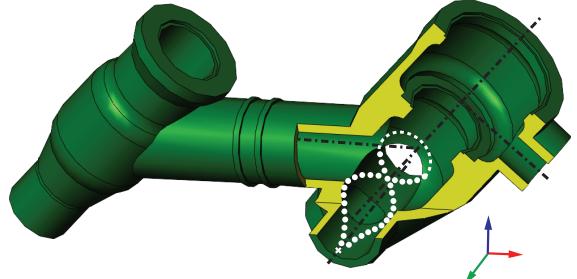
(a) Picture of the real environment



(b) Experiment #1



(c) Experiment #2



(d) Experiment #3

Figure 18: Experiment description. a) Picture of the real facility available for testing. b-d) 3D CAD models (cut views) and paths of the three experiments which were carried out. The start positions of the trajectories are shown on the environment picture and by crosses on the CAD models.

For these experiments, the robot was remote controlled by a human operator and 3D scans were taken approximately every 0.1m. The path lengths and number of 3D scans of each trajectory are respectively 5.8m (59 scans), 4.3m (37 scans) and 4.2m (42 scans). Each experiment was performed once.

### 5.3 3D scan registration characterization

The scan registration method (ICP) presented in Section 4.4 is then applied on these scan data sets. Because the test structure has several openings, the external surroundings of the test environment can be in the field of view of the 3D scanner. In order to be in real operating conditions (only one opening to place the robot in the structure), all points which do not belong to the inner casing of the structure were removed (manually filtered). The points belonging to the robot body are also automatically removed, because they remain static in the scanner frame.

### 5.3.1 Point cloud down-sampling optimization

The down-sampling of the 3D point clouds has a strong influence on scan matching. Actually the ICP computation time that is  $O(N \log M)$  when using kD-trees can be drastically reduced by decreasing the number of points to process. In order to use the localization tool online, it is then important to keep the computation time low, while not losing precision. The down-sampling rate ( $\lambda$ ) was then optimized for the given application.

The optimization is done using the two last scans of the third experiment, because they were taken at close positions and have a high overlap. Since there is no exact ground truth available, the ground truth is defined as the result of scan matching using all the points ( $\lambda = 1$ ). The translation error  $|\mathbf{t}(\lambda_{GT} = 1) - \mathbf{t}(\lambda)|$  between both 3D transformations  $T(\lambda_{GT} = 1)$  and  $T(\lambda)$  is used to measure the error induced by down-sampling. The product between the computation time and the translation error is used as optimization criterion. Figure 19 summarizes this optimization process. The scan registration was repeated 16 times for different down-sampling rates from 0.5% to 20%. The error distributions are reported in box plots on the left-hand figure. Their weighted median values are fit with an exponential function represented by the dashed curve and the 95% confidence interval. Figure 19 only shows sub-sampling rates from 0.5% to 4% but the fit was done with percentages up to 20%. From 4% all trials are in the confidence interval. The optimization criterion (for  $N = M = 341,500$  points) is shown on the right figure. The minimum of the confidence interval indicates that randomly selecting 2.5% of the points is a good compromise to reduce computation time and not loose precision. The results presented hereafter are then generated with this sub-sampling rate.

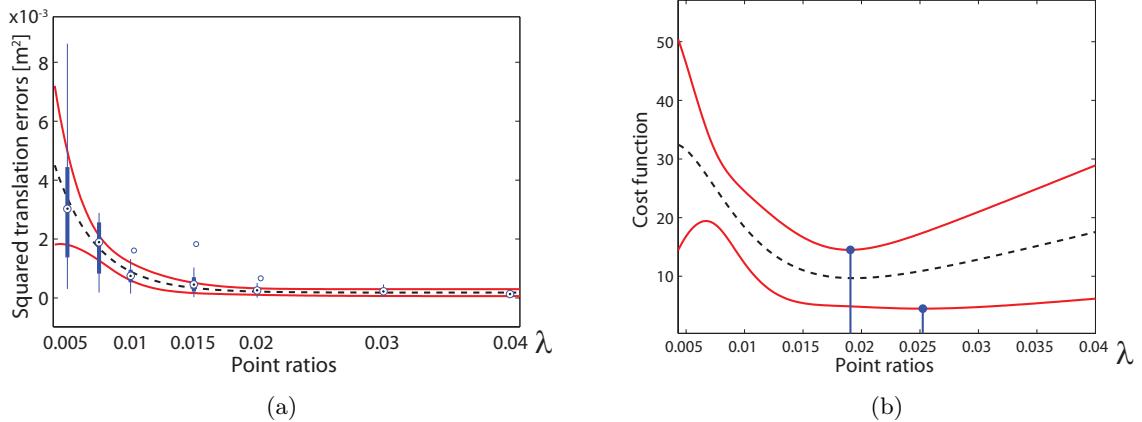


Figure 19: Scan matching: down-sampling rate optimization. a) Squared translation error vs. down-sampling rate. The ground truth is given by matching both scans without down-sampling ( $\lambda = 1$ ). b) Cost function defined as the product of computation time and translation error. The two last scans of the third experiment were used to perform this optimization.

### 5.3.2 3D mapping

Figure 20 shows the result of the registration applied to the 42 scans of the third data set with 2.5% down-sampling. The grey dots represent the sensor positions (frame H) where 3D scans were taken. Similarly, Figure 2 and 22 show the reconstruction of the first and second experiments. By looking in more detail at these figures, it can be noted that there is a dark track along the scanner positions. These are actually due to points belonging to the robot structure which were not perfectly filtered. Furthermore, even if 3D scans are correctly aligned some outlier points appear below the main pipe in Figure 2. These points taken by the last scans of the experiment illustrate the lower precision of the sensor when the distance and incidence angle are high as pointed out in Section 3.2.

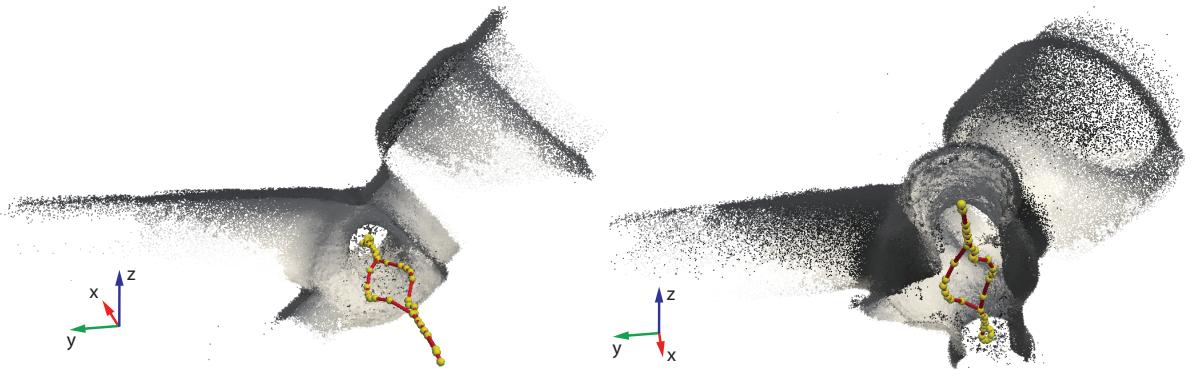


Figure 20: Result of 3D scan matching and environment reconstruction of the third experiment (cut views). Yellow dots indicate the scan positions in the sensor frame H. Left: side view. Right: bottom view. 2.5% of points are used for scan matching and for this reconstruction visualization.

These reconstructions qualitatively show that ICP allows us to correctly align scans and to build consistent 3D maps. A quantitative evaluation is however difficult, because there is no exact ground truth available for experiments carried out in such complex environments. We can take benefit from the loop closing of the third experiment to compare the global registration from scan 1 to scan N ( $T_{1N} = T_{12} \cdot T_{23} \cdot \dots \cdot T_{N-1N}$ ) to the limited ground truth, respectively the registration between scan 1 and N ( $T_{1N_0}$ ) evaluated with a high percentage of points ( $\lambda = 1$ ). This evaluation leads to a translation error of 20mm for a traveled distance of 4.2m. This result gives an insight in the error accumulated by ICP. The error is actually depending on several parameters such as the range finder noise, the type of environment, the traveled distance between scans (overlap) as well as the down-sampling rate.

### 5.3.3 Limitations

One should not forget that the environment should contain enough features in the 3D scanner measurement range and field of view to constrain the scan matching. The MagneBike platform is well suited for the specified environment. Nevertheless if it had to work in environments with long straight cylindrical sections, it should differently combine the sources of information. It should actually rely only on 3D odometry in these parts and use 3D scanning

for visualization, but not for localization. A limit case is presented in Figure 7, the robot stands in a rather homogeneous pipe, but can still observe enough features (*e.g.* diameter changes, tube intersections) to constrain the scans registration as shown in the reconstruction (Figure 2). The featureless cylindrical parts should approximately not be longer than 2m.

#### 5.4 3D odometry characterization

Similar to 3D registration, there is no exact ground truth to evaluate 3D odometry. Because we know from the previous section that scan matching is very precise (the translation error on the third data set when using 20% of the points is actually smaller than 20mm for a 4.2m path length) the N-1 relative transformations between the N scans of that experiment are used as a limited ground truth for the evaluation.

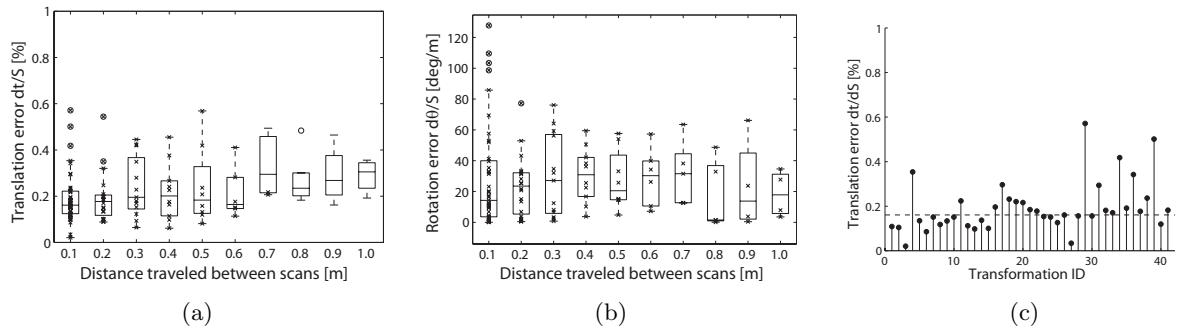


Figure 21: 3D odometry error for the third experiment. a-b) Translation  $|t_{\text{ICP},ij} - t_{\text{ODO},ij}|/S_{ij}$  and rotation  $|\theta_{\text{ICP},ij} - \theta_{\text{ODO},ij}|/S_{ij}$  errors relative to the traveled distance for different scan intervals (0.1 to 1m). c) Detail of the translation errors for 0.1m scan interval.

3D odometry is hence evaluated by comparing the relative transformations between scanning positions  $i$  and  $j$  given by ICP ( $T_{\text{ICP},ij}$ ) and 3D odometry ( $T_{\text{ODO},ij}$ ). To reduce the quantity of information and ease comprehension of the results, the six DoF transformations are reduced to a rotation angle  $\theta$  and the amplitude of motion (translation)  $t$ . The error metrics are defined as the rotation and translation errors relative to the path length  $S_{ij}$ , *i.e.*  $|\theta_{\text{ICP},ij} - \theta_{\text{ODO},ij}|/S_{ij}$  and  $|t_{\text{ICP},ij} - t_{\text{ODO},ij}|/S_{ij}$ .

Figure 21 shows the translation (a) and rotational (b) errors as a statistic over the whole trajectory. The statistic is given for different scanning intervals. Each box plot analyzes all the relative transformations between the scans with indices  $i$  and  $i + 1$ ,  $i$  and  $i + 2$ , ... The path length is then approximately 0.1m for the first box plot, 0.2m for the second one and so on until 1m. Box plots are used to represent the error distributions that are not normal. Errors are actually defined positive and their distribution is asymmetric with most of the errors small and some outliers (high errors). The median value is then an appropriate descriptor of the error distributions, respectively representing the error with the highest probability. The graphs show that the translation error is around 15% (0.15m per meter traveled) and rotation error around 25° per meter traveled. Graph 21(c) shows the detail on the translation error for an average traveled distance of 0.1m between scans.

## 5.5 3D odometry combined with scan matching

3D odometry and scan matching results are combined as explained in Section 4.5. Figure 22 (left and center) for instance shows the result of the second experiment, *i.e.* the inner casing reconstruction (point cloud), the robot trajectory in the robot and sensor frames (in blue and red) as well as the scan positions in the sensor frame H (yellow spheres). The right plot shows the same fused trajectory (in black) and the one computed with pure 3D odometry that slowly drifts (in grey).

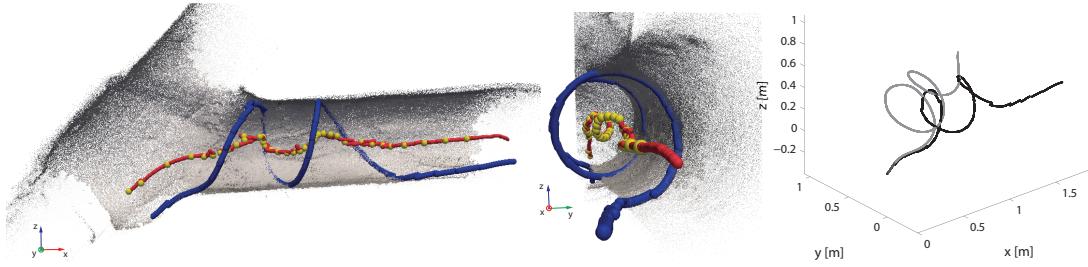


Figure 22: Left and center: 3D point cloud reconstruction (2.5% of points) of the second experiment with the robot trajectory in the robot and sensor frames (blue and red), the scan positions are marked by yellow dots. Right: 3D trajectory with pure odometry (grey) and combination with the scan matching results (black).

## 5.6 Discussion

### 5.6.1 3D odometry as initial guess for scan registration

Given the good performance of 3D odometry, the 3D movement of the robot can be used as a prior input for the 3D scan registration. This has several advantages. It firstly improves the computation time, since less iterations are necessary for the registration. Furthermore, a good initial guess will decrease the probability that ICP converges to the wrong local minimum, thus the combination makes the localization procedure quite robust. Finally, knowing that 3D scanning is the slowest process of the localization procedure (50s for a high resolution 3D scan), a good initial guess allows us to optimize the distance traveled between 3D scans and consequently decrease the total inspection time.

To analyze the relevance of the initial guess, the scan registration performance with and without the 3D odometry initial guess are compared. The third experiment is used again, because it allows us to utilize the best limited ground truth and contains the most complex combination of motions. The ground truth is evaluated by matching the scans with 20% of down-sampling. The performance is measured by observing the percentage of ICP failures, respectively when the translation error of 3D scan matching with the ground truth position differs from more than 0.02m. The translation error is actually very small ( $<0.02\text{m}$ ) if ICP finds the correct local minimum and high if it diverges. Similar to the 3D odometry

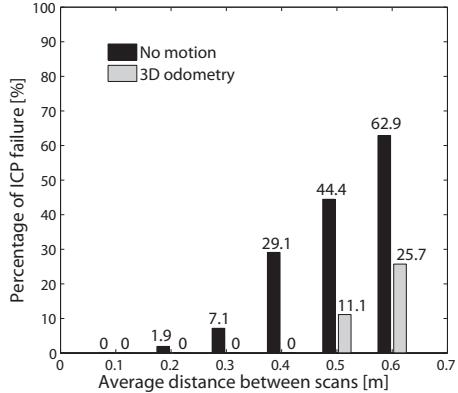


Figure 23: 3D odometry as initial guess for ICP. Percentage of scan pair registrations that failed with ("3D odometry") and without ("No motion") initial guess.

evaluation (Section 5.4), this experiment is repeated for different distances between the 3D scan positions. The results are reported in Figure 23. They show that for small displacements (0.1m), 3D odometry is not absolutely necessary since the algorithm matches 100% of the scans. From 0.2m, there are already some cases for which ICP is not able to match pairs of scans when no prior input is provided to the algorithm. From distances larger than 0.5m, both methods have outliers, which means that either the odometry guess is no longer sufficient or that the scan overlap is too low for ICP to converge.

### 5.6.2 Design guidelines for optimization

The MagneBike localization concept having been characterized in detail, one can deduce a few guidelines to optimize the localization procedure regarding reliability and time:

- **Distance between scans:** as presented previously getting a 3D scan approximately every 0.4m of travel is a good compromise to ensure a sufficient overlap for ICP and to reliably use 3D odometry as an initial guess. Furthermore, 3D odometry is precise enough to track the robot trajectory between the scan positions.
- **Environment curvature and shape:** the previous guideline should however not only be based on the distance traveled by the wheels, but also on the environment shape that influences the system behavior. More precisely, for a given trajectory the distances traveled by the robot and sensor frames depend on the surface curvature as depicted in Figure 24. A small robot displacement can lead to a large rotation (*e.g.* surface with high curvature) and even a large sensor displacement on convex surfaces (*e.g.* second phase of step climbing). A large rotation may strongly influence the sensor field of view and consequently decrease the scan overlap, especially in small diameter tubes. Thus the criterion to decide when a 3D scan has to be taken should not only be based on the traveled distance, but also on the robot rotation.
- **Point cloud density:** given that a down-sampling rate of 2.5% is sufficient to reliably match 3D scans, it is possible to increase the sensor rotational speed and consequently

proportionally decrease the scanning time. This should however be done in the limits of the sensor actuation mechanics. The current implementation allows to provide a maximum rotation velocity of 3rpm (limited by the actuator). It is then possible to decrease the density to 20%, respectively getting a full 3D scan in 10s. One should also consider the scan resolution that is necessary for the environment reconstruction. This constraint mainly depends on the usage of the final map. 2.5% is sufficient for visualization as shown in Figures 2, 20 and 22, nevertheless it may be necessary to get higher resolution scans for obstacle detection (*e.g.* very small holes in the structure) or for path planning purposes. A compromise to balance the time spent for moving and localization is to decrease the point cloud density to 25% of full resolution. This enables to get a scan in 12.5s which is approximately half the time that is necessary to move from 0.4m at 1m/min.

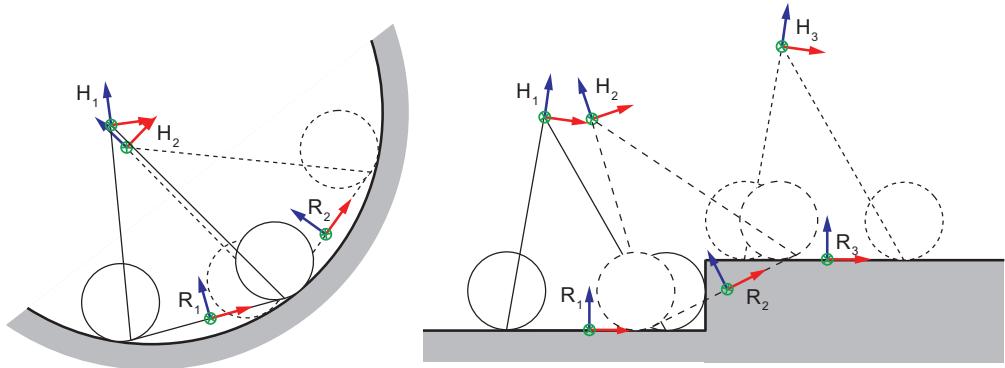


Figure 24: Comparison of the movement of the robot (R) and sensor (H) frames depending on the surface curvature.

### 5.6.3 3D localization accuracy

To conclude this discussion, the performance of the localization procedure is compared to the objectives defined in Section 2.1:

- To achieve a position accuracy of  $\pm 5\text{cm}$  required for safe remote operation, results presented in Sections 5.1 and 5.4 (*i.e.* translation error of 15%) show that an error of 5cm is accumulated after a traveled distance of 0.33m. Ensuring this accuracy requires to take a 3D scan every 0.33m. This constraint is another input to the optimization guidelines proposed in the previous section.
- Regarding orientation accuracy, one can draw similar conclusions. Hence, with a rotation error rate of  $25^\circ/\text{m}$ , presented in Section 5.4, an accuracy of  $\pm 10^\circ$  is ensured if the robot travels distances up to 0.4m between 3D scans.
- The  $\pm 1\text{cm}$  accuracy required for defects localization is however more critical, because 1cm of error is already accumulated after 7cm of travel. To ensure 1cm accuracy, it is then recommended to refine the robot position by acquiring a 3D scan when a defect is detected by inspection sensors.

## 6 Conclusion

The MagneBike robot is an innovative climbing robot with high mobility developed for inspecting complex shape steel structures (Tâche et al., 2009), especially steam chests environments which are used as a specific case study. This type of environment is challenging for the locomotion concept (*e.g.* climbing ability, obstacle passing), but also in regarding to localization. The goal is to get the position of the robot on the 3D surface (six DoF), as well as to provide a 3D model of the environment that is *a priori* unknown. These tools are necessary to control the robot. They can either be used as 3D visualization for the user who remote controls the robot that is hidden in its confined environment or as input information for autonomous control purposes (*e.g.* obstacle detection, active mechanism control or path planning).

This paper presents a localization strategy based on 3D scan registration and 3D odometry. A 3D scanner enables the acquisition of local 3D maps of the environment. Consecutive scans are subsequently registered using the iterative closest point (ICP) algorithm and utilized to build a global 3D map. The position between the 3D scans is tracked by 3D odometry which is also used as prior guess for the scan registration. The paper describes the design and implementation of the 3D scanner which must be compact and lightweight to satisfy the available space and payload constraints. The paper also presents the 3D odometry approach which is based only on wheel encoders and a three axis accelerometer. Because the robot moves slowly, the accelerometer data can be used to estimate the surface inclination with respect to gravity. To compensate for the limited sensory inputs, the odometry model includes an estimation of the local surface curvature.

3D localization was implemented on the MagneBike and field tests were performed in order to evaluate and characterize the system in real application conditions. The tests conclude that the 3D point clouds taken by the 3D scanner contain enough features to reliably register scans. Moreover, 3D odometry is precise enough to track the robot position between scans and provides a reliable input for the scan registration. The limitations were also identified. For instance, the odometry model suffers from a singularity, the robot cannot track its position when it follows sideways paths with the gravity vector staying exactly parallel to the accelerometer axis. In this paper, we do not only provide a characterization and a validation of the localization process, but also propose some guidelines to optimize it. It is actually possible to optimally down-sample the 3D scan to maintain the registration precision high while drastically decreasing the computation time. The scan positions can also be optimally distributed to decrease the amount of time dedicated to scanning, since the robot must stand still when scanning.

## 7 Future Work

This work describes the tools and input information that are necessary to provide a visual feedback to the operator. The future work then consists in building a graphical interface with enhanced features that help the user to interpret the 3D environment. The point clouds should be converted into meshes or geometric surfaces for better rendering. Edges could for

instance be highlighted to better visualize step obstacles that require special control of the active lifter mechanism. Other obstacles on which the robot cannot drive (*e.g.* holes, ridges) should be detected and displayed. All these features are not only useful for visualization purpose, but also necessary inputs to automate the robot control. The system could for instance automatically detect steps, control the lifters and pass over the obstacle. Based on the guidelines described in this paper, the robot could automatically decide when it is necessary to take a 3D scan to ensure the correct convergence and optimize the inspection time. An interesting feature would be to identify long cylinders that cannot be reliably matched by ICP and instead rely on odometry in these sections. Once all these goals have been reached, the platform will be ready for the next testing phase, respectively its deployment in similar environments that are only available during outage time.

Regarding long term objective of autonomous inspection, the system should be able to autonomously plan its path along the complex 3D surface. It should identify the environment traversability (*e.g.* obstacle to avoid, obstacle that can be overcome, surface inclination, ...) in order to plan the optimal trajectory towards the goal position. Once these improvements will be achieved for a single robot, a future goal is to extend the inspection system to a team of several robots that can independently achieve tasks by cooperation as described in (Breitenmoser et al., 2010). Multi agent systems allow for faster execution of tasks, increased robustness and flexibility due to parallelism, redundancy and decentralization.

## Acknowledgments

The authors would like to thank Laurent Kneip for its contribution to the implementation of the 3D range finder and the control electronic for the MagneBike, Rudolph Triebel for the constructive discussions about the 3D localization and mapping problems as well as ALSTOM Power Service for supporting this work as exploration research. This work is supported by Swiss CTI project 8439.1 "Highly Compact Robots for Power Plant Inspection".

## References

- Ahrary, A. (2009). Localization of a mobile robot in sewer pipes using hough transform. *ICGST International Journal on Graphics, Vision and Image Processing*, 9(1):17–23.
- Ahrary, A., Tian, L., Kamata, S.-I., and Ishikawa, M. (2007). Navigation of an autonomous sewer inspection robot based on stereo camera images and laser scanner data. *International Journal on Artificial Intelligence Tools*, 16(4):611–625.
- Alwan, M., Wagner, M. B., Wasson, G., and Sheth, P. (2005). Characterization of infrared range-finder PBS-03JN for 2-D mapping. In *Proc. of the 2005 IEEE International Conference on Robotics and Automation (ICRA'2005)*, pages 3947–3952, Barcelona, Spain.
- Beeson, P., O’Quin, J., Gillan, B., Nimmagadda, T., Ristroph, M., Li, D., and Stone, P. (2008). Multiagent interactions in urban driving. *Journal of Physical Agents: Multi-Robot Systems*, 2(1):15–29.
- Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.

- Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.
- Bonnifait, P. and Garcia, G. (1999). 6 DOF dynamic localization of an outdoor mobile robot. *Control Engineering Practice*, 7(3):383–390.
- Breitenmoser, A., Tâche, F., Caprari, G., Siegwart, R., and Moser, R. (2010). MagneBike - toward multi climbing robots for power plant inspection. In *Proc. of the 9<sup>th</sup> International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2010)*.
- Hertzberg, J. and Kirchner, F. (1996). Landmark-based autonomous navigation in sewer-age pipes. In *Proc. of the First Euromicro Workshop on Advanced Mobile Robot (EUROBOT'1996)*, pages 68–73, Kaiserslautern, Germany.
- Inuktun (2009). Inuktun. <http://www.inuktun.com>.
- Kawata, H., Ohya, A., Yuta, S., Santosh, W., and Mori, T. (2005). Development of ultra-small lightweight optical range sensor system. In *Proc. of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2005)*, pages 1078–1083, Edmonton, Canada.
- Kneip, L., Tâche, F., Caprari, G., and Siegwart, R. (2009). Characterization of the compact Hokuyo URG-04LX 2D laser range scanner. In *Proc. of the 2009 IEEE International Conference on Robotics and Automation (ICRA'2009)*, pages 1447–1454, Kobe, Japan.
- Lamon, P. (2005). *3D Position Tracking for All-Terrain Robots*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL).
- Morris, A., Ferguson, D., Omohundro, Z., Bradley, D., Silver, D., Baker, C., Thayer, S., Whittaker, C., and Whittaker, W. (2006). Recent developments in subterranean robotics. *Journal of Field Robotics*, 23(1):35–57.
- Nuechter, A., Surmann, H., Lingemann, K., and Hertzberg, J. (2003). Consistent 3D model construction with autonomous mobile robots. In *Proc. of the 26th Annual German Conference on Artificial Intelligence (KI'03)*, pages 550–564, Hamburg, Germany.
- Okubo, Y., Ye, C., and Borenstein, J. (2009). Characterization of the Hokuyo URG-04LX laser rangefinder for mobile robot obstacle negotiation. In *Proc. of the Society of Photo-Optical Instrumentation Engineers SPIE*.
- Pomerleau, F., Colas, F., Ferland, F., and Michaud, F. (2009). Relative motion threshold for rejection in ICP registration. In *Proc. of the 7th International Conference on Field and Service Robotics (FSR'09)*, Cambridge, Massachusetts, USA.
- Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the ICP algorithm. In *Proc. of the Third International Conference on 3D Digital Imaging and Modeling (3DIM'2001)*, pages 145–152, Quebec City, Canada.
- Stone, W. C., Juberts, M., Dagalakis, N., Stone, J., and Gorman, J. (2004). Performance analysis of next-generation LADAR for manufacturing, construction, and mobility. Technical report, National Institute of Standards and Technology, Gaithersburg, Maryland, USA.
- Surmann, H., Lingemann, K., Nüchter, A., and Hertzberg, J. (2001). A 3D laser range finder for autonomous mobile robots. In *Proc. of the 32nd International Symposium on Robotics (ISR'2001)*, pages 153–158, Seoul, Korea.

- Suzumori, K., Miyagawa, T., Kimura, M., and Hasegawa, Y. (1999). Micro inspection robot for 1-in pipes. *IEEE/ASME Transactions on Mechatronics*, 4(3):286–292.
- Tâche, F., Fischer, W., Caprari, G., Moser, R., Mondada, F., and Siegwart, R. (2009). Magnebike: A magnetic wheeled robot with high mobility for inspecting complex shaped structures. *Journal of Field Robotics*, 26(5):453–476.
- Thrun, S., Thayer, S., Whittaker, W., Baker, C., Burgard, W., Ferguson, D., Haehnel, D., Montemerlo, M., Morris, A., Omohundro, Z., Reverte, C., and Whittaker, W. (2004). Autonomous exploration and mapping of abandoned mines. *IEEE Robotics and Automation Magazine*, 11(4):79–91.
- Ueda, T., Kawata, H., Tomizawa, T., Ohya, A., and Yuta, S. (2006). Mobile SOKUIKI sensor system: Accurate range data mapping system with sensor motion. In *Proc. of the 3rd International Conference on Autonomous Robots and Agents (ICARA'2006)*, Palmerston North, New Zealand.
- Weingarten, J. W., Gruener, G., and Siegwart, R. (2004). A state-of-the-art 3D sensor for robot navigation. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS'2004)*, pages 2155–2160, Sendai, Japan.
- Wisspeintner, T., Maurelli, F., Droeschel, D., May, S., Surmann, H., and Pervoelz, K. (2008). A 3D laser scanner system for intersection analysis and autonomous driving. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS'2008)*, Nice, France.
- Wulf, O. and Wagner, B. (2003). Fast 3D scanning methods for laser measurement systems. In *Proc. of the 14th International Conference on Control Systems and Computer Science (CSCS'2003)*, Bucharest, Romania.
- Ye, C. and Borenstein, J. (2002). Characterization of a 2-D laser scanner for mobile robot obstacle negotiation. In *Proc. of the 2002 IEEE International Conference on Intelligent Robotics and Automation (ICRA'02)*, pages 2512–2518, Washington DC, USA.
- Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152.