sciendo

Slovak Journal of Civil Engineering

Vol. 30, 2022, No. 2, 47 – 58
DOI: 10.2478/sjce-2022-0014

LIDAR-BASED MOBILE MAPPING SYSTEM FOR AN INDOOR ENVIRONMENT

Ján BRINDZA[1], Pavol KAJÁNEK[1*], Ján ERDÉLYI[1]

Abstract

The article deals with developing and testing a low-cost measuring system for simultaneous localisation and mapping (SLAM) in an indoor environment. The measuring system consists of three orthogonally-placed 2D lidars, a robotic platform with two wheel speed sensors, and an inertial measuring unit (IMU). The paper describes the data processing model used for both the estimation of the trajectory of SLAM and the creation of a 3D model of the environment based on the estimated trajectory of the SLAM. The main problem of SLAM usage is the accumulation of errors caused by the imperfect transformation of two scans into each other. The data processing developed includes an automatic evaluation and correction of the slope of the lidar. Furthermore, during the calculation of the trajectory, a repeatedly traversed area is identified (loop closure), which enables the optimisation of the trajectory determined. The system was tested in the indoor environment of the Faculty of Civil Engineering of the Slovak University of Technology in Bratislava.

Address

[1] Department of Surveying, Slovak University of Technology, Bratislava, Slovakia

[*] ***Corresponding author:*** pavol.kajanek@stuba.sk

Key words

- Lidar,
- IMU,
- Wheel speed sensor,
- SLAM,
- 3D model,
- Loop closure,
- Trajectory optimisation,
- Scan matching.

1 INTRODUCTION

The article deals with designing and testing a low-cost measuring system for simultaneous localisation and mapping (SLAM) using lidar. SLAM can be applied indoors and outdoors, in water, the air, and underground for the navigation of autonomous systems, mapping purposes, and the geometric control of objects. The mobile measuring system consists of three orthogonally-placed 2D lidars, an inertial measuring unit (IMU), a stereo camera, and wheel speed sensors. The data processing model of the system enables the calculation of the system's trajectory and the creation of a 3D model of the object measured. Due to the continual movement of the robot, the position and orientation of the lidars change. Especially during the acceleration and deceleration of the system, the inclination values rise, which significantly affect the polar coordinates measured by the lidars and generate errors in the 3D model. In addition, an error in the system's orientation causes a rotation in the model. Errors accumulated in this way eliminate the detection of loop closures (Chen, S. et al., 2021; Cummins, M., Newman, P. 2007). The use of a loop closure refines the resulting model and the trajectory travelled (Stachniss, C. 2009).

Identifying the repeatedly visited area (the loop closure) in the data processing suppresses the impact of any systematic errors on the resulting model and optimises the trajectory and model based on this information. However, finding a repeatedly traversed environment and subsequently optimising the trajectory is also difficult. The article describes a comprehensive data processing model, which increases the model's accuracy based on the alignment of the scans, the use of the loop closure, and the optimisation of the trajectory. The proposed algorithm was tested by experimental measurements, and the results were subsequently analysed.

The map created by SLAM is based on the lidar scans measured and the actual position of the measuring system. The data captured in different positions must be transformed into one common coordinate system. The process of referencing the measured data, which includes calculating the transformation parameters, is called scan matching.

Based on the sensors used, SLAMs are divided into lidar-based SLAMs (Biström, B., 2019), camera-based SLAMs

© 2022 The Author(s). This is an open access article licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License (http://creativecommons.Org/licenses/by-nc-nd/3.0/).

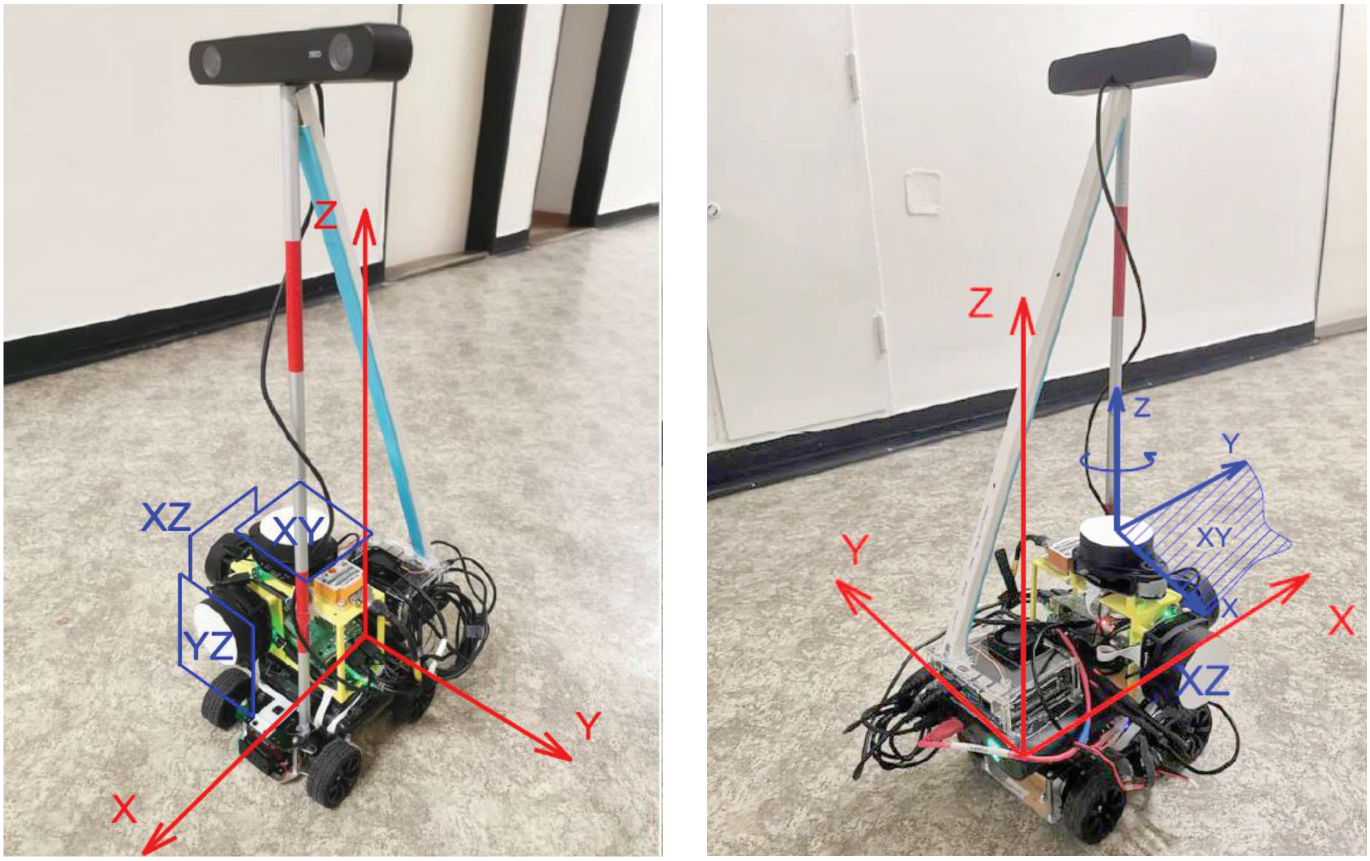


Fig. 1 *Measuring system developed with a marked coordinate system and the position of the lidars*

(visual SLAMs) (Cummins, M., Newman, P., 2007), and combinations thereof.

The lidar-based approach (Chen, S. et al., 2021; Gutmann, J. S., Konolige, K. 1999) uses lidar sensors that measure directions and lengths based on a laser beam that rotates at a constant angular velocity. After the reflection of the laser beam from the surrounding environment, the direction and the length are evaluated; then the coordinates of the points is determined. The results of the measurement are a 2D point cloud in the lidar coordinate system, where the points are placed at the intersection of the plane (on which the lidar rotates) with the surrounding environment. As the measuring system moves, the position and orientation of the coordinate system of the lidar change, and these parameters can be expressed with a transformation of the point clouds measured. The robot's displacement is represented by translation, and the change in orientation is defined by rotation.

Stereoscopic or monocular cameras are used in the visual SLAM (Thrun, S., et al., 2003; Valencia, R., Andrade C., J., 2018; Prokhorov, D. et al., 2019), and a point cloud is created from the images based on identical points. However, it is a visual system, which has a problem with a less textured surface (such as white walls). Other issues include blurred images caused by the system's movement and shaking or the sun's rays reflected from shiny surfaces onto the camera lens.

In multisensory systems (Gutmann, J. S., Konolige, K., 1999), lidar is usually combined with other sensors such as IMU, global navigation satellite systems (GNSS), wheel speed sensors, and cameras. The combination of lidar with another sensor is used most often because lidar data processing (Chen, S. et al., 2021) has lower computing power than the processing data from a camera. Therefore, it is easier to implement real-time SLAM, which is not affected by ambient lighting conditions. However, high-end lidars are expensive, have a robust construction and high-power requirements, and are unsuitable for mobile devices. In addition, many commercial systems use GNSS for localisation, but their application is limited to outdoor use. The proposed approach enables the application of SLAM in indoor areas where no GNSS signal is available.

All lidar scans are referenced into a single coordinate system, which is usually called scan matching. Scan matching can be divided into:

1. Element-based scan matching minimises the distance between the elements of two scans. The transformation parameters can be calculated based on features (points or lines) (Biström, B., 2019; Fernando, M. et al. 2014).
2. The graph-based approach uses graphs, which contain nodes that represent the measured position of the robot; these nodes are connected by lines that define the dependencies between the individual positions of the robot (Kudriashov, A. et al., 2020; Grisetti, G. et al. 2010; Lu, F., Milios, E. 1997).
3. The probabilistic approach is based on the extended Kalman filter (Skrzypczyński, P., 2009; Montemerlo, M., Thrun, S., 2007; Kümmerle, R. et al., 2009).

Commercial systems are very expensive, i.e., on the order of 10,000 to 100,000 euros, making them impractical for some applications. Therefore, the focus of the research was on the design of a low-cost system (Tab. 1) and the testing of its capabilities.

2 CHARACTERISTICS OF THE MEASURING SYSTEM

The design measuring system is built on an Ackerman robotic platform, which means that the rear axle is fixed and the front axle rotates when the system changes direction. The measuring system

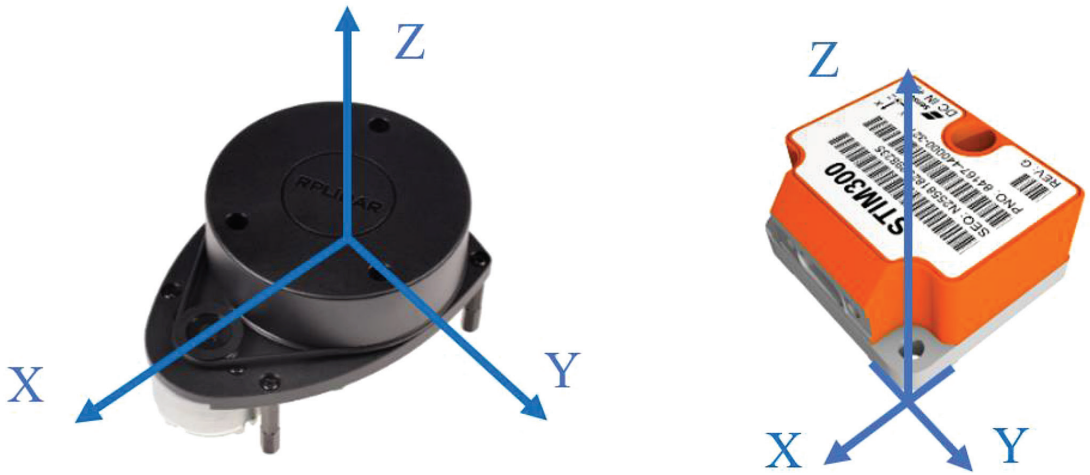


Fig. 2 Lidar frame (left) and IMU frame (right)

consists of three orthogonally-placed 2D lidars, IMUs, a stereo camera, and wheel speed sensors. The control of all the sensors is resolved by the robot operating system (ROS).

Three coordinate systems are defined:

- The reference coordinate system expresses the position and orientation of the system, the trajectory of the motion, and the final model.
- The coordinate system of the measuring system, which fuses the measurements from all the individual sensors,
- The sensor coordinate system, in which the data are captured.

The coordinate system of the measuring system is orthogonal (Fig. 1 – red line); the Y-axis passes through the rear axle with the wheel speed sensors installed. The X-axis is perpendicular to the Y-axis and passes through the longitudinal axis of the platform in the direction of the system's movements. The Z-axis is perpendicular to the X and Y axes and points vertically upwards. The coordinate system of the sensors is defined by the manufacturers of each sensor. The lidar coordinate system (Fig. 2 on the left – blue line) has the origin defined on the lidar's rotation axis. The X-axis passes through the rotation axis of the lidar and the motor's rotation axis that provides the lidar's rotation. The Z-axis is represented by the upward rotation axis of the lidar, and the Y-axis is completed by an orthogonal right-handed coordinate system (Grisetti, G. et al., 2010). The lidars are marked based on the plane on which the measurements were realized (Fig.1 - blue plane). The IMU coordinate system (Fig. 2 on the right) has its origin formed by the intersection of the X and Y axes. The Y-axis is given by the longitudinal edge of the IMU on the connector side. The Z-axis is perpendicular to the X and Y axes and creates a right-handed coordinate system (Lu, F. - Milios, E., 1997).

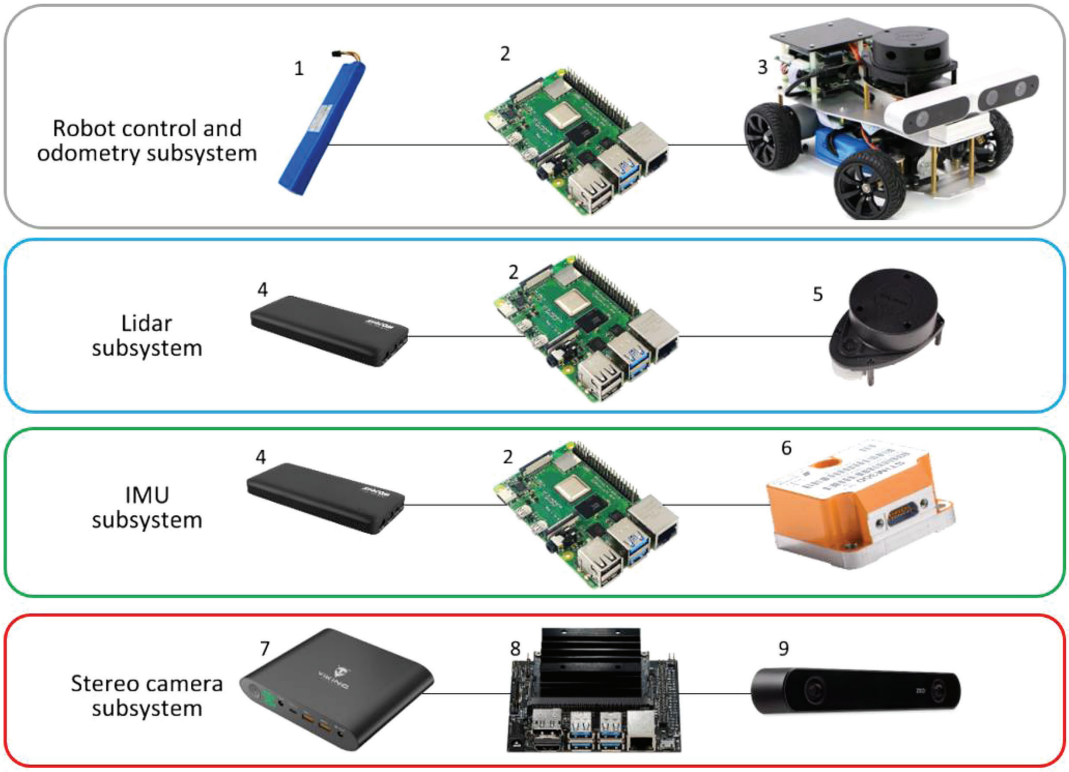


Fig. 3 Topology of the measuring system

The sensor part of the system is divided into four subsystems (Fig.1), namely:

1. Robot control and odometry,
2. The lidar,
3. IMU,
4. Stereo camera.

The whole system (Fig. 3) is managed by the ROS (the ROS version is Melodic), where the subsystems represent nodes that communicate with each other via a Wi-Fi network. ROS provides for the synchronisation and cooperation of the individual sensors. The master microcontroller NVIDIA Jetson Nano 4GB (Fig. 3 - 8) (NVIDIA Autonomous Machines, 2021) creates ROS because it has the highest computing performance. Raspberry Pi 4 8GB (Fig. 3 - 2) microcontrollers are used for the lidar and IMU. All the microcontrollers are represented in the ROS as nodes, and each node is identified by its IP address. All the microcontrollers have the Ubuntu 18 operating system. The system's initialisation (ROS initialisation, definition of measurement nodes and parameters), measurement storage, and the start/end of the measurements are performed via remote access on the main microcontroller. The data measured are downloaded through a Wi-Fi network. The measurements are stored in a database, where each measurement is represented as a message. Each message contains the time of the measurement and information about the coordinate system. In the future, the system will allow us to control the movement of the platform based on the data processed and thus create a fully autonomous system.

The movement of the system is performed using a robotic platform (Fig. 3 - 3), which is controlled by a Raspberry Pi 4 microcontroller (Fig. 3 - 2) (Raspberry Pi Trading Ltd., 2019) with an expansion module (STM32F103RBT6). This module is used for remote control of the robot motors and storage data from the wheel speed sensors. It also contains an additional IMU and magnetometer.

The Raspberry Pi 4 8GB microcontroller (Fig. 3 - 2) is connected to an expansion module (STM32F103RBT6), which also has an IMU and a magnetometer, which serves as the remote control of the motors. The control subsystem is powered by a 12V lithium polymer battery (Fig. 3 - 1). The rear axle is fixed, and the front axle allows the wheels to rotate and change direction. While a pair of motors drives the wheels on the rear axle, the incremental wheel speed sensors are installed on the shaft of both wheels. The platform's direction is calculated using the difference between the increments captured by the wheel speed sensors and the distance between the wheels. A servo motor also controls the platform direction.

The lidar subsystem consists of three 2D Slamtec RPlidar A1 (Fig. 3 - 5), placed on three perpendicular planes on a common frame. The Slamtec RPlidar A1 is a 2D 360° laser scanner with a range of 12 m and performs measurements with a frequency of 6 Hz (Shanghai Slamtec CO., 2020). The three lidars are controlled by a Raspberry Pi 4 8GB microcontroller (Fig. 3 - 2), and the subsystem is powered by a 5V 3A battery (Fig. 3 - 4). The whole subsystem is connected to the ROS based on the IP address of the microcontroller.

The IMU subsystem consists of the IMU Sensonor STIM 300 (Fig. 3 - 6) (Sensonor AS, 2013), which is connected to the Raspberry Pi 4 8GB microcontroller (Fig. 3 - 2), which connects the entire system to the ROS via Wi-Fi. The whole system is powered by a 5V 3A battery (Fig. 3 - 4). The sampling frequency for inertial measurements is 200 Hz.

ZED 2 (Fig. 3 - 9) (Camera Z.E.D., 2021) is a stereo camera located approximately 60 cm above the robot platform. It is connected to the NVIDIA Jetson Nano 4GB (Fig. 3 - 8). The whole subsystem is powered by a 12V 4A power supply (Fig. 3 - 7). Due to the extensive data produced, a more powerful motherboard is used for the stereo camera. The microcontroller also contains a module for the Wi-Fi to connect to the ROS.

Tab. 1 *Budget for the measuring system*

Components of the measuring system	Cost
NVIDIA Jetson Nano 4GB microcontroller	143 €
3x Raspberry Pi 4 microcontroller	257 €
Stereo camera ZED 2	505 €
3x Slamtec RPlidar A1 lidar	375 €
IMU Sensonor STIM 300	3,251 €
robotic platform	405 €
power supply 12V 4A	94 €
2x battery 5V 3A	30 €
12V lithium polymer battery	25 €
Total:	**5,085 €**

3 DESIGN OF THE DATA PROCESSING MODEL

The mathematical data processing model combines measurements from the lidar, the IMU, and the wheel speed sensors. The main problem is the possible lower accuracy of the final model of the surrounding environment due to the accumulation of errors in the position and orientation of the system. Measurements from the stereo camera have not yet been used in the current version of the data processing model.

The data processing model (Fig. 1) consists of the following steps:

1. Data loading and conversion (from message to matrix form),
2. Lidar pre-processing,
3. Estimation and elimination of the lidar's inclinations,
4. Scan matching,
5. Loop closure,
6. Optimisation of the trajectory and model.

The data measured from all the subsystems are stored in a bag file, where the measurements are recorded as messages with the time and information about the sensor's coordinate system. Polar coordinates represent the measured points on the lidar scans. The data should be filtered (by length) because the system contains structural elements that generate noise in the resulting model. The minimum distance is set to 0.5 m for the filtering and the maximum distance to 6 m.

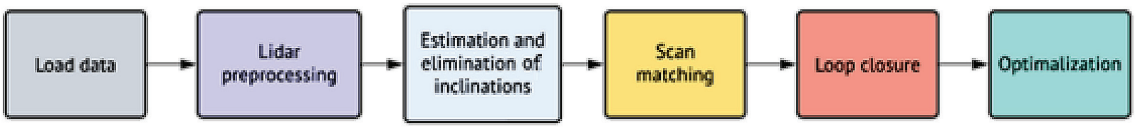


Fig. 4 *Data processing scheme*

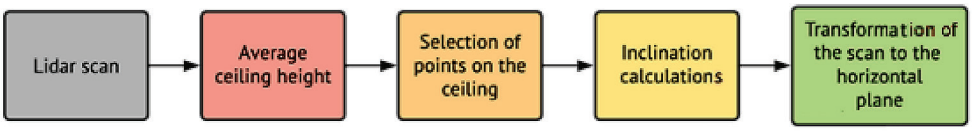


Fig. 5 Scheme for the elimination of slopes on the XY plane

3.1 Estimation and elimination of the inclination of the lidar

The orientation of the lidars changes as the system moves, especially as it accelerates and decelerates and due to any uneven floor. This tilt of the lidar also affects the polar coordinates (changes in lengths). The problem is resolved by the transformation of the scans onto a horizontal plane. The determination of the inclination assumes that the three lidars are placed on a common frame and that their mutual orientation does not therefore change. The tilt of the lidar on the XY plane is determined by the measurements in the XZ and YZ vertical sections (Fig. 1). For the estimation of the platform tilt, it is assumed that the system is moving in an indoor environment and that the room's ceiling is horizontal.

The tilt of the lidar is determined as the tilt of the regression line is determined from the points on the ceiling. Therefore, it was decided to automatically select the points in each scan belonging to the roof and then calculate the platform's tilt, following the scheme in Fig. 5.

3.1.1 Detection of points on the ceiling

When detecting points on the ceiling, we assumed that the points on the roof had the highest height. However, objects on the top (such as lamps) negatively affect the estimation of the tilt, so iterative calculations were used. In the first step, the regression line was estimated based on the points selected using the average height of the ceiling. Next, the average height of the roof was determined from the set of the highest points in the scan. In the second step, the regression line parameters were estimated based on the points selected 0.05 m from the regression line calculated in the first step. The regression line calculated should faithfully represent the line that belongs to the roof's plane (Fig. 6). Finally, the platform's tilt could be determined using the direction of the regression line determined.

There may be a situation where the calculations do not select any point on the roof due to a poorly determined average height; the resulting regression line will therefore have zero coefficients. Therefore, the algorithm uses a condition for the minimum height of the ceiling above the lidar, defined by the regression line's coefficient (offset). If the offset of the regression line is less than 1.5 m, the slope value of the previous scan will be used. The value of the regression line offset is verified based on the average height of the room. The second condition used in the algorithm defines the maximum change in the tilt. If the difference in the tilt between the previous and current step (scan) is greater than 1.2° for the YZ section and 3° for the XZ section, the algorithm uses the value estimated in the previous state of the system. Different thresholds have been used because any change in the tilt in the direction of the movement (XZ) will be greater than in the direction perpendicular to the motion (YZ). The accuracy of the inclination determined only by IMU decreases with the measurement time, which is caused by the accumulation of errors in the integration of the angular velocity.

The scans on the XY plane are finally transformed into a horizontal plane using the inclination (tilt) calculated on the XZ and YZ planes. Next, the alignment of the scans on the XY plane was performed using a similar transformation, where the rotation matrix was calculated from the estimated slopes in the XZ and YZ planes. Since the lidar scan is two-dimensional, a zero Z coordinate was added. After the transformation, the Z coordinates were removed.

3.2 Scan matching

An essential part of the data processing is the scan matching. This step determines the rotation and translations between two consecutive scans that correspond to the system's movement. This step aims

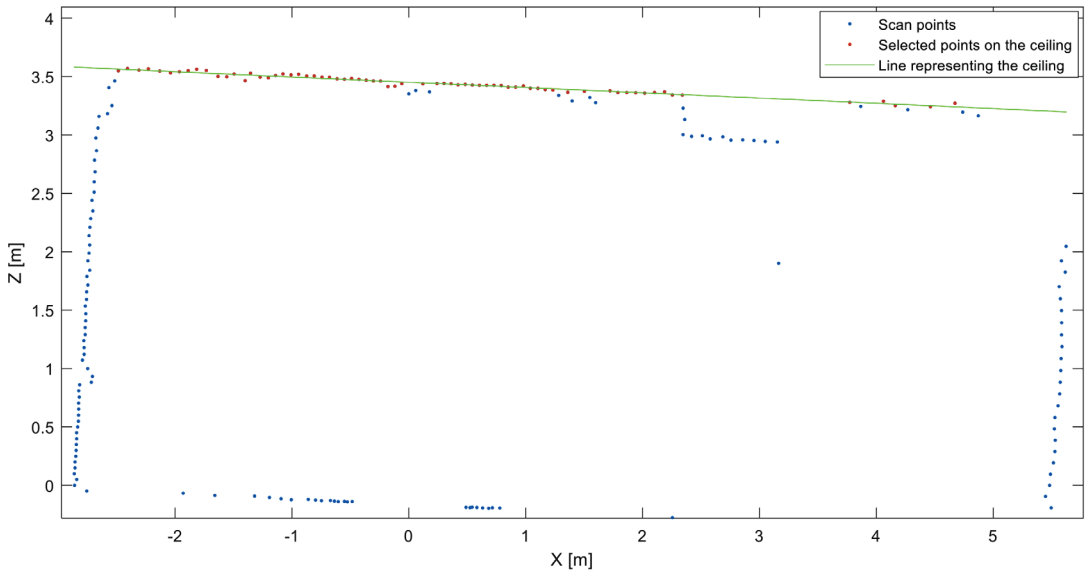


Fig. 6 Estimation of the platform tilt from the lidar measurements

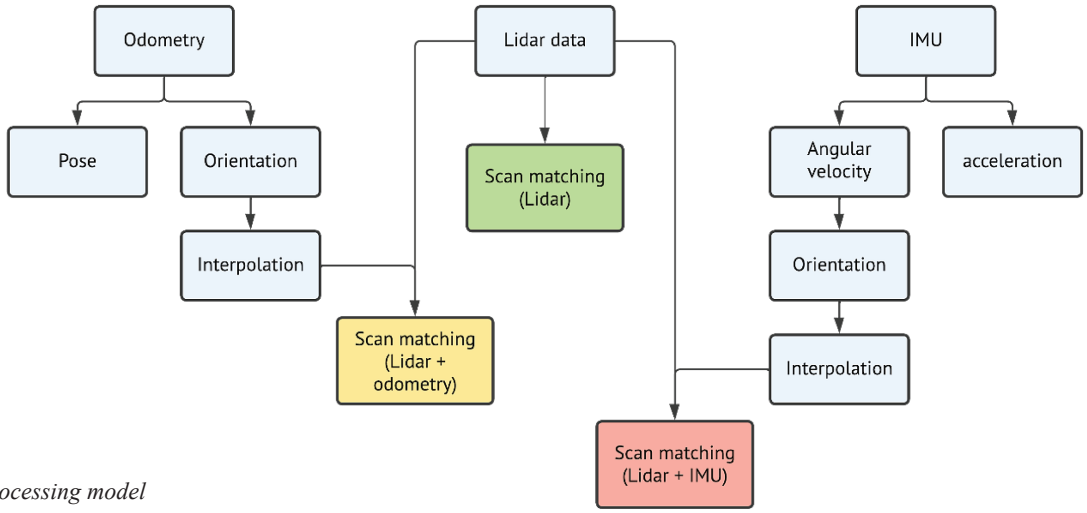


Fig. 7 Data processing model

to determine the translation and rotation of the current scan on the reference (previous) scan. The normal distribution transformation determines the transformation parameters, and the initial parameters for the normal distribution transformation were determined based on the correlation. The initial position using the correlation is determined by comparing two scans in which the current scan moves and rotates over the reference scan until the highest degree of correlation is identified. Next, the scan is converted to a grid, and the centre of gravity is determined for each cell before the transformation. The transformation parameters are determined from the coordinates of the centre of gravity. Finally, the transformation parameters calculated are used to transform the scan into a common floor-plan model.

The final product should be a 3D model of the scanned environment. The main goal is to create a 2D model of the floor plan with a good trajectory and orientation first and then add the scans on the XZ and YZ planes to create a 3D model. The following approaches are used for the creation of the floor plan model (Fig. 7):

1. Scan matching – using only lidar scans,
2. Combination of scan matching and odometry (orientation used from the odometry),
3. Combination of the scan matching and IMU (direction used from IMU).

The current scan was transformed into the previous (reference) scan using scan matching. The initial transformation parameters (translation and rotation) are determined based on the correlation between the scans. The final estimate of the transformation parameters is determined based on the normal distribution transformation (Biber, P. - Straßer, W., 2003). Before the transformation, a grid with a cell size of 0.05 m is created from both scans. For each grid cell, the centre of gravity is calculated from the coordinates of the points located in the cell. The centres are used to calculate the transformation parameters (translation and rotation). The transformation parameters are the change in orientation and the translation between the previous and current scans, i.e., the relative orientation and relative position of the system. From the relative position, the absolute position is calculated based on the absolute position of the previous scan. The absolute position of the robot in the reference coordinate system defines the origin of the lidar coordinate system in which the lidar measurements are realised. Based on the results of the scan matching, all the scans are transformed into a reference coordinate system, in which a floor plan and model are created.

The scan matching was supplemented by the correction of the transformation function parameters based on the wheel speed sen-

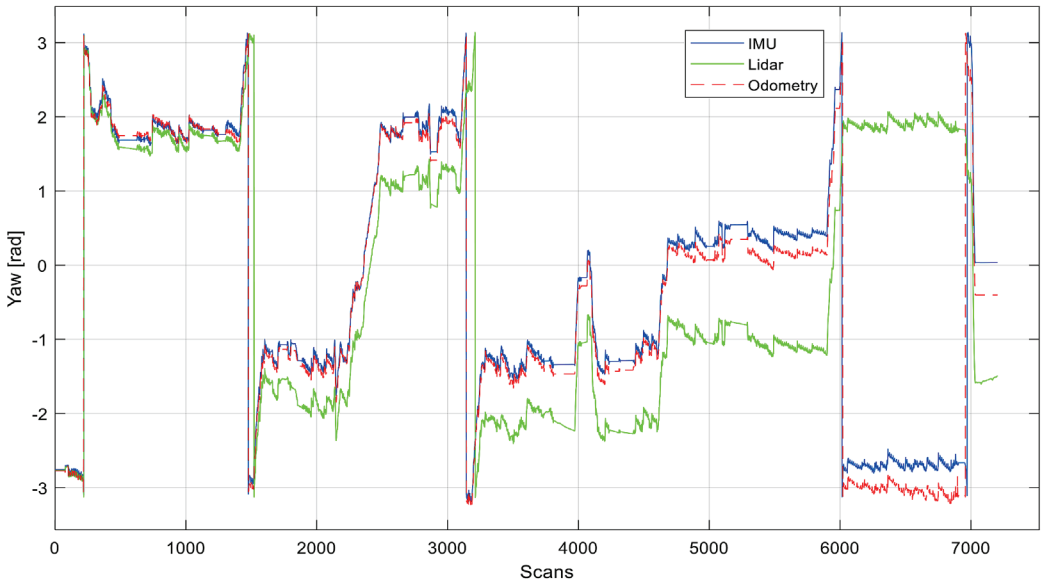


Fig. 8 Comparison of the orientation calculated from the IMU, Lidar and odometry

sors and IMU measurements. Due to the different sampling frequencies of these sensors, all the measurements were interpolated to the lidar time stamp (Fig. 8).

The absolute direction given by the scan matching was corrected using the orientation determined by the odometry (the second approach) and subsequently using the orientation given by the IMU (the third approach). Based on the estimated position and orientation of the lidars, a model (floor plan) was created.

3.3Loop closure

The loop closure is performed within the scan matching, where the repeated environment is identified (Fig. 9). The principle of loop closure identification is based on comparing the currently measured lidar scan with the database of the measured scans and the best match. Only scans up to 3 m from the current scan are selected from the database of the measured scans. In the case of multiple selected scans, the earliest match scan will be marked as a loop closure scan.

The loop closure is a part of the scan matching, and it is used to identify the repeated environment (Fig. 9). The principle of loop closure identification is to compare the current lidar scan with a database of the measured scans to find the best match. The database of measured scans only contains scans up to 3 m from the current scan. The position of the scans is stored in the trajectory of the system. The loop closure scan is the earliest scan in the database whose match is higher than the threshold.

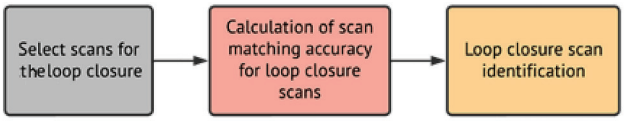


Fig. 9 *Scheme of loop closure*

The threshold is set adaptively based on the environment, while the algorithm automatically detects a more complex environment. In a complex environment, the threshold for loop closure detection is reduced, and the quality of the scan matching obtained from the correlation is used for the best results. A smaller search radius of the potential scans is also used for the loop closure in a complex environment.

3.4 Optimalization of the trajectory

The optimisation of the trajectory is a part of the scan matching. It is the process that occurs when the loop closure is identified. The trajectory between the current scan and the loop closure scan is corrected during the optimisation. The corrections are a function of the time.

The procedure corrects the trajectory in both the ΔX, ΔY position and the Δα orientation. The error in the ΔX, ΔY position and the Δα orientation (1) is expressed as the difference between the current robot position X_0, Y_0, α_0 (Fig. 10, on the left side) and the position determined by the transformation of the current scan into a loop closure X_P, Y_P, α_P (Fig. 10 on the right side).

$$\Delta X = X_P - X_0, \quad \Delta Y = Y_P - Y_0 \quad \Delta\alpha = \alpha_P - \alpha_0 \tag{1}$$

First, we considered that the SLAM error is the result of the accumulation of sensor errors that accumulate during the measurement. Therefore, we assume that the error in position v_i^x, v_i^y and orientation v_i^α depends on the time of measurement t. Therefore, the error in the position and orientation of scan i at the time t_i are calculated from the optimisation coefficients k_x, k_y, k_α and time t_i, which elapsed from the loop closure scan to the currently optimised scan.

$$v_i^x = k_x \cdot t_i, \quad v_i^y = k_y \cdot t_i \quad v_i^\alpha = k_\alpha \cdot t_i \tag{2}$$

The optimisation coefficients k_x, k_y, k_α (Equation 3) are calculated as the ratio of the error in position ΔX, ΔY and the orientation $\Delta\alpha$ with the time Δt (Equation 4) that has elapsed between the loop closure scan measured at the time t_n and the actual scan measured at the time t_m.

$$k_x = \frac{\Delta X}{\Delta t}, \qquad k_y = \frac{\Delta Y}{\Delta t} \quad \text{a} \quad k_\alpha = \frac{\Delta\alpha}{\Delta t} \tag{3}$$

$$\Delta t = t_m - t_n \tag{4}$$

The accuracy of SLAM is also influenced by other factors such as the surrounding environment (long straight corridors are problematic for processing the data) or a rapid change in the orientation of the system. On the other hand, a controlled change in the orientation of the lidar on the measuring system can be used to refine the model.

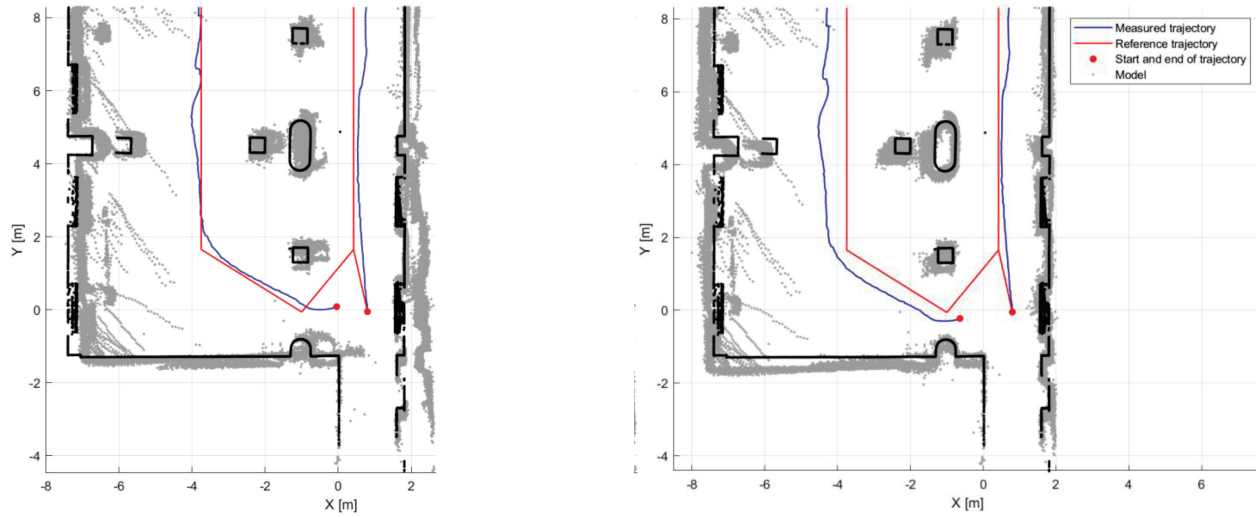


Fig. 10 *Trajectory and model calculated without any optimisation (on the left side) and with optimisation (on the right side)*

4 TESTING THE MEASURING SYSTEM

This part of the article deals with the test of the low-cost measurement system developed, which is described in Chapter 2. The trajectory of the robotic platform and floor plan model was calculated from the data measured. Finally, the results were compared with a reference model created that is based on an independent measurement from the Trimble TX5 terrestrial laser scanner.

The scanning resolution was set at 15 mm/10m, and the measurement of a single point was repeated 3 times. The distances between the scanner and the scanned object ranged from 1.5 m to a maximum of 6 m in each case. Then the average density of the point cloud was approximately 3 mm (represents the distance between two neighboring points of the point cloud on the surface of the scanned object). With the use of the mentioned scanner and considering the measurement conditions stated, the accuracy of a single measured point was less than 2 mm in any case. The scanning was performed from 5 stations of the scanner. During the processing of the data measured, the point clouds from several of these stations were registered into a common coordinate system for an evaluation of the results from the measuring system developed. The overlap between the point clouds from two adjacent stations was always above 60%.

The experiment was done in a classroom at the Faculty of Civil Engineering, Slovak Technical University in Bratislava. First, the trajectory (Fig. 11) was predefined as a closed path that passes through a complex environment. The trajectory has the shape of a rectangle with dimensions of 27 m x 4.7 m. Part of the environment contains a lot of furniture (benches), which is problematic for a correct estimate of the transformation. The next part of the trajectory has a simple geometry (straight walls) that contains only a few elements for a proper estimation of the translations.

The predefined trajectory (Fig. 11) begins at the starting point S. The system moves around the perimeter of the room, and the measurement ends at point E so that the trajectories contain the repeated environment. The average speed of the measuring system during the experiment was 0.4 $m \cdot s^{-1}$. The total length of time of the measurement was 4 minutes. The system was remotely controlled during the experiment.

The processing of the measurements was performed by the model described above. It was possible to evaluate the benefits of the individual methods for obtaining the accuracy of the system's position, length, orientation and the resulting model.

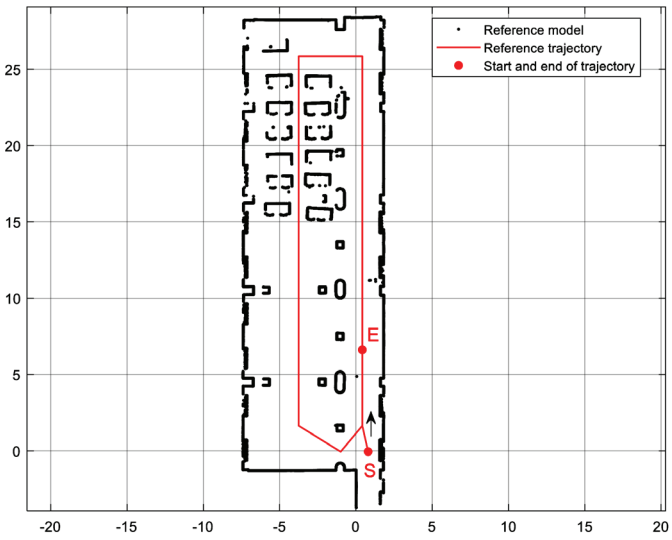


Fig. 11 Scheme of the room and the predefined trajectory

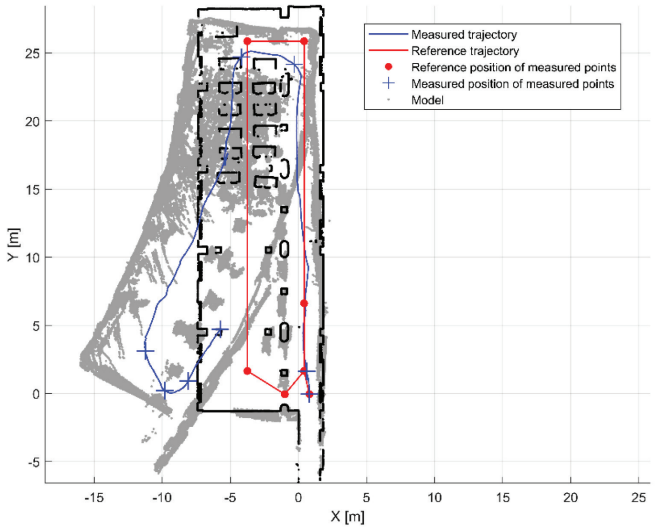


***Fig. 12**: Trajectory and model calculated only with scan matching from the lidar scans*

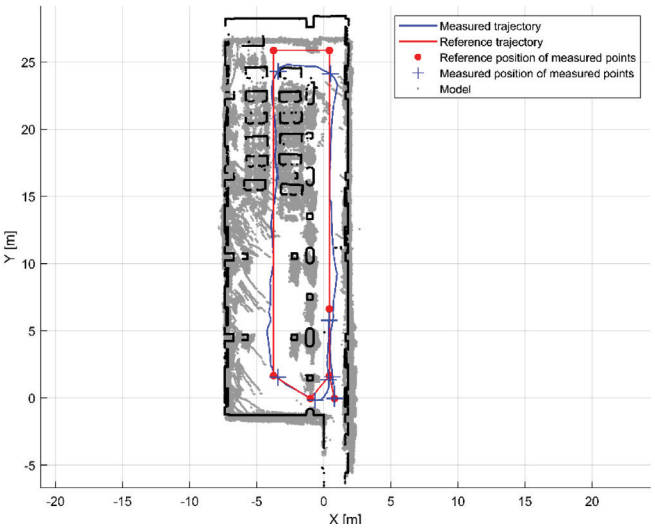


***Fig. 13** Trajectory and model calculated based on the combination of scan matching and odometry*

4.1 Results of the experimental measurements

We use only a scan matching approach without eliminating any system inclinations in the first step. The result is very noisy, and the model is inconsistent (Fig. 12). In addition, an error in orientation occurs when using only scan matching (Fig. 12). When correcting the direction of the system by orientation from the odometry measurements, we increase the accuracy of the trajectory and the accuracy of the model of the surrounding environment (Fig. 13).

In the next step, the elimination of lidar slopes was used in the scan matching. Then, the inclination of the lidars was automatically evaluated based on the points lying on the ceiling of the room. The resulting model should be less noisy than before the elimination of the inclination, which can be seen from the scan matching based on the lidar scan's elimination of the platform's inclination (Fig. 14) compared to the results from the data processing without the elimination of the inclination (Fig. 12).

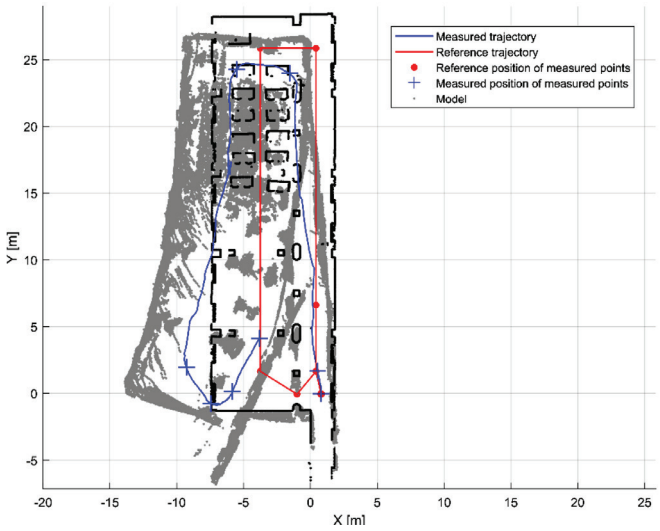


Fig. 14 Trajectory and model calculated based on scan matching from the lidar scans with the elimination of the lidar inclination

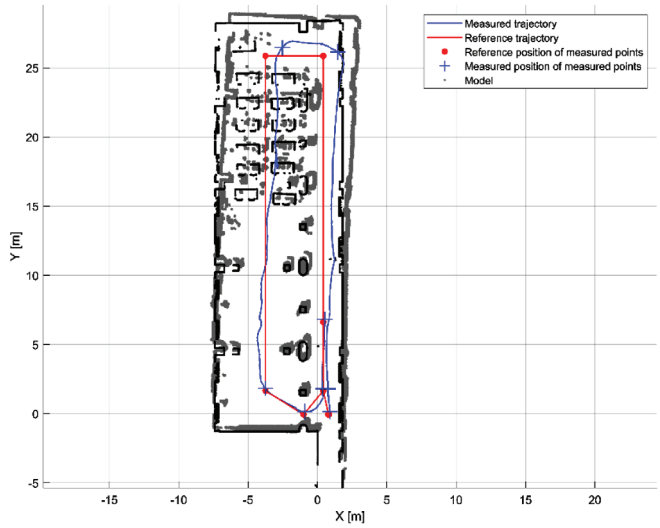


Fig. 15 Final model and trajectory calculated based on scan matching with the loop closure and optimisation of the trajectory

In the calculations with optimisation, a loop was continuously searched for, i.e., it finds a repeatedly traversed environment. Based on the identification of the loop closure, the algorithm transforms the current scan into an already created model of the traversed environment (Fig. 15).

5 ANALYSIS OF THE EXPERIMENTAL MEASUREMENTS

The analysis of the results enables us to evaluate the accuracy of the position and orientation of the proposed system. The individual approaches to eliminating errors are compared based on the differences in the position, length, and orientation. The differences are calculated by comparing the measured values with the reference values.

The positional deviations were calculated at the measured points (MP), representing the trajectory's location where the orientation changed significantly. The positional differences Δp were determined using the reference position MP X_{ref}, Y_{ref} and the measured position MP X_{means}, Y_{means} from each scan matching approach.

$$\Delta p = \sqrt{(X_{ref} - X_{meas})^2 + (Y_{ref} - Y_{meas})^2} \tag{5}$$

Tab. 2 *Comparison of approaches to elimination of error based on the positional difference*

Approach	The average difference in the position [m]	The maximum difference in the position [m]
Scan matching	4.3	8.8
Scan matching + odometry	0.7	1.7
Scan matching with the elimination of slopes	3.6	6.5
Scan matching with loop closure	0.3	1.0

The worst results were achieved by approaching only the scan matching, where the average difference in position is 4.3 m and the differences increase up to 8.8 m. The increased difference in the position (Fig. 16) is mainly caused by the long straight sections of the classroom, which are not suitable for the transformation of scans.

A combination of the scan matching with the odometry significantly reduced the average difference in position to 0.7m, representing a 76% improvement. Furthermore, scan matching with the elimination of the inclination has an average deviation at the 3.6 m position and a maximum deviation at the 6.5 m position. Finally, scan matching with a loop closure and optimisation

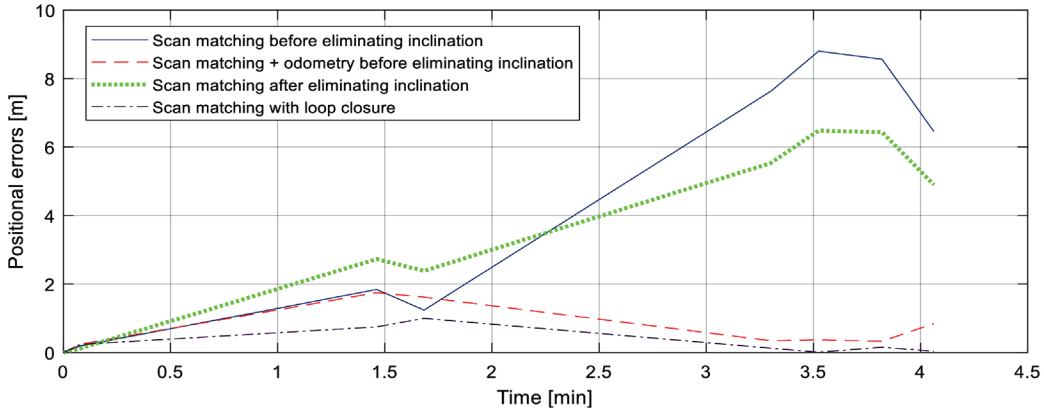


Fig. 16 Comparison of approaches to elimination of error based on the positional difference

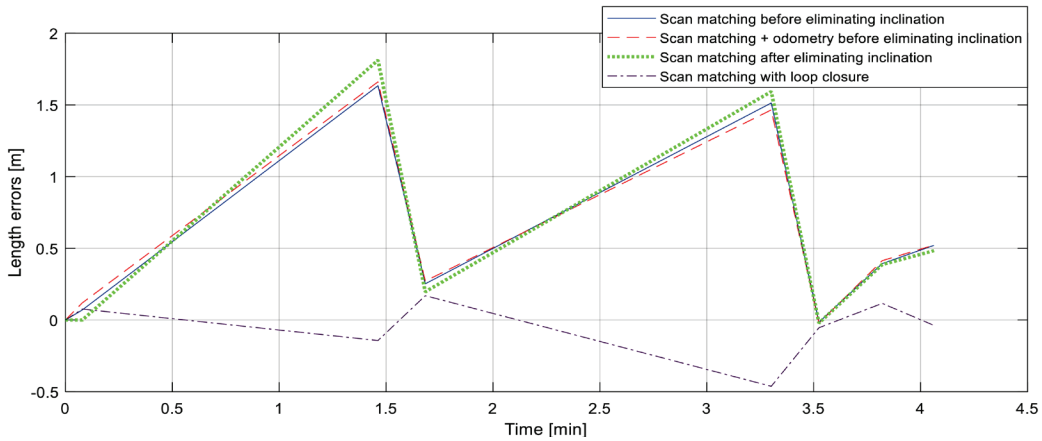


Fig. 17 Comparison of the approaches to elimination of error based on the difference in length

achieved the best degree of accuracy. The average difference in position is 0.3 m, and the maximum difference is 1.0 m.

In the second step of the analysis, the differences in length between the measured points were evaluated. The differences in length were calculated as the difference between the reference and measured length. The measured length was calculated from the coordinates of the measured points (MP) from the individual approaches.

$$\Delta\alpha = \alpha_{ref} - \alpha_{meas} \tag{6}$$

Tab. 3 *Comparison of the approaches to elimination of error based on the difference in length*

Approach	The average difference in length [m]	Maximum length difference in length [m]
Scan matching	0.6	1.6
Scan matching + odometry	0.6	1.7
Scan matching with the elimination of slopes	0.6	1.8
Scan matching with loop closure	0.2	-0.5

Based on the difference in length (Fig. 17), we can see that the most significant length errors are achieved on the long straight sections, which are problematic for scan matching. In the first three approaches, the difference in length has the same average value of 0.6 m, while the maximum length error is 1.8 m. The most minor length errors are achieved by scan matching with the loop closure (Tab. 3), where the average difference in length is 0.2 m, and the maximum difference in length is -0.5 m, which is calculated for the long straight section.

The accuracy of the orientation was evaluated based on the differences in orientation $\Delta\alpha$ on the straight lines of the points measured. The difference in orientation $\Delta\alpha$ was calculated as the difference between the reference orientation α_{ref} and the measured orientation α_{means}.

In the analysis of the values calculated based on the degree of accuracy in the orientation, the errors in orientation on the direct MP lines were calculated. The orientation error $\Delta\alpha$ was calculated as the difference between the reference orientation α_{ref} and the measured orientation α_{means}. The reference orientation α_{ref} was calculated from the reference coordinates of the measured points (MP). The measured orientation α_{means} was calculated from the measured coordinates of MP from the individual approaches.

$$\Delta\alpha = \alpha_{ref} - \alpha_{meas} \tag{7}$$

The orientation of the system determined in the first approach (scan matching) and the third approach (scan matching with the elimination of the lidar inclinations) has a low degree of accura-

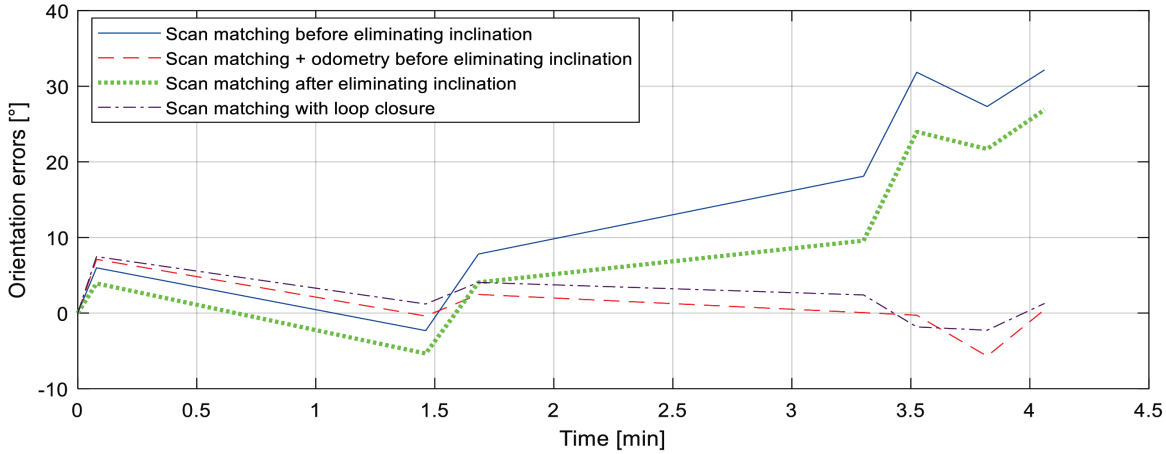


Fig. 18 *Comparison of approaches to the elimination of errors based on the differences in orientation*

cy (Fig. 18). The average difference in orientation for an approach with only scan matching is 18°. The approach with the elimination of lidar inclinations reduces the average difference to 13.6°. The best result for the orientation was achieved by scan matching with a correction by orientation from the odometry. The average difference in orientation for this approach is 2.4°. The final approach with the loop closure had an average difference in orientation of 2.9° (Tab. 4).

Tab. 4 *Comparison of approaches to eliminating errors based on the difference in orientation*

Approach	The average difference in orientation [°]	The maximum difference in orientation [°]
Scan matching	18.0	32.2
Scan matching + odometry	2.4	7.1
Scan matching with the elimination of slopes	13.6	26.9
Scan matching with loop closure	2.9	7.5

The best results in the differences in position and length were achieved in the final approach with the loop closure and optimisation of the trajectory and the model. The scan matching with the corrected orientation from the odometry achieved the highest degree of accuracy of the orientation, but the difference between the second approach (scan matching and the odometry) and the final approach with the loop closure is minimal.

A different model was calculated to better evaluate the accuracy of the proposed system and the processing model. The difference model represents the differences in the point positions between the reference model and the model calculated by scan matching with the loop closure and optimisation.

The difference model shows that 44% of the points have a difference in position up to 0.10 m (Fig. 19, green points); 21% of the points have a difference in a range from 0.10 m to 0.20 m; 13% of the points have a difference in a range from 0.20 m to 0.40 m, and 22% of the points have a difference in position of more than 0.40 m.

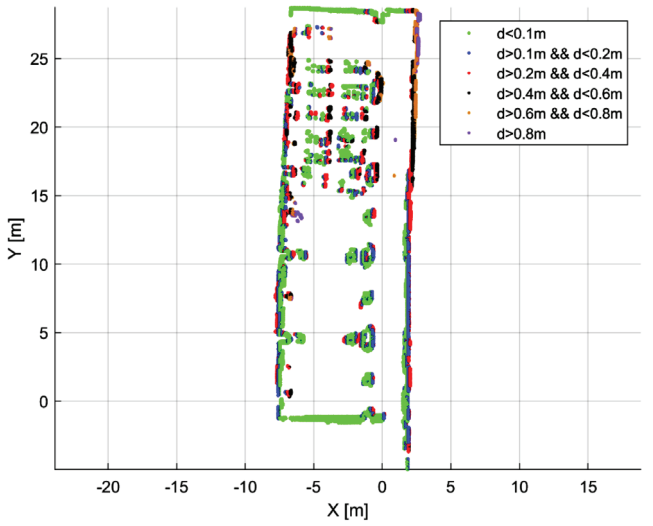


Fig. 19 *Difference model*

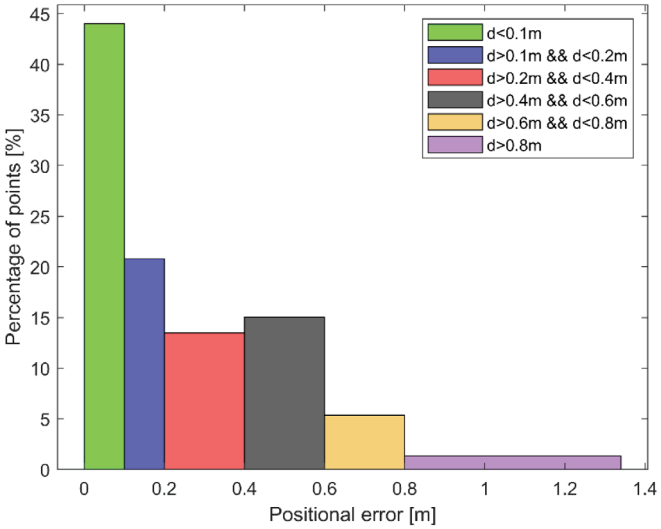


Fig. 20 *Histogram of the differences in position from the difference model*

6 CONCLUSION

Due to the high cost of existing mobile mapping systems, the work deals with designing a low-cost system consisting of three 2D lidars, an IMU, and wheel speed sensors. All the sensors are controlled by a robotic operating system (ROS), which ensures the connection of several sensors into one autonomous mapping system. The article focuses on the application of SLAM using the mentioned sensors. During the research, a low-cost measuring system was developed that consists of several sensors used to eliminate errors and increase the accuracy of the trajectory and the model created. The resulting trajectory was calculated based on the sensors used, and a model (floor plan) was created based on the trajectory. The main part of the data processing was the scan matching, where the transformation between two consecutive lidar scans was calculated based on identical points.

The model for processing the measurements of all the sensors is presented in the article. The results, which represent an estimate of the trajectory system, are used to create a model of the surrounding environment. During the data processing, errors accumulate with the length of time of the measurement, and the accuracy of the system decreases. For this reason, the main aim of the processing was to effectively suppress systematic errors in the position and orientation of the system and thus improve the accuracy of the final model.

The proposed processing model contains an algorithm for automatically identifying lidar inclinations and their elimination. In addition, there is an algorithm for identifying loop closures in the next processing step, i.e., repeatedly visited places. Based on the loop closure, the trajectory between the pair of scans that identify the same part of the environment is optimised.

The accuracy of the position and orientation of the proposed system was evaluated based on the known coordinates of the points measured from a predefined trajectory. We assessed the system's degree of accuracy based on a differential model, which was calculated as the difference between the reference model and the final model using the proposed processing model. The classroom reference model was determined by independent terrestrial laser scanning. When calculating the model using measurements, individual solutions were used to individually evaluate the effect of the individual methods on the suppression of errors in the po-

sition and orientation. We performed the analysis based on the difference in position, length, and orientation, which were determined from the measured points from a predefined trajectory.

Based on the comparison of methods used in the model created, eliminating lidar inclinations improves the resulting floor plan model. The loop closure and optimisation significantly refine the trajectory as well as the model. The data processing was enhanced by combining the lidar measurements with other sensors such as encoders (odometry) because the odometry has a better long-term degree of accuracy in the orientation. The difference model shows that 44% of the points have a difference in position of up to 0.10 m.

In our work to come, we want to focus on the application of the Kalman filter in estimating the position and orientation of the system. The Kalman filter will make it possible to optimise the estimation of the system's state based on predictions and current measurements while considering the accuracy of the measurements. This approach should improve the system's accuracy in downloaded situations for estimating the transformation between a pair of lidar scans (long straight corridors). We also want to use a stereo camera and code marks to refine the results.

Acknowledgement

This work was supported by the Slovak Research and Development Agency under Contract No. APVV-18-0247.

REFERENCES

Biber, P. - Straßer, W. (2003) *The normal distributions transform: A new approach to laser scan matching*. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003). IEEE, 2003. pp. 2743-2748.

Biström, B. (2019) *Comparative analysis of properties of LiDAR-based point clouds versus camera-based point clouds for 3D reconstruction using SLAM algorithms*. 2019.

Camera, Z. E. D. (2021) SDK Overview. Available online: Stereolabs. com/assets/datasheets/zed2-camera-datasheet. pdf (accessed on 16 Aug. 2021).

Chen, S., et al. (2021) *A LiDAR/Visual SLAM Backend with Loop Closure Detection and Graph Optimization*. Remote Sensing, 2021, Vol. 13, No. 14, p. 2720.

Cummins, M. - Newman, P. (2007) *Probabilistic appearance based navigation and loop closing*. In: Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE 2007, pp. 2042-2048.

Fernando, M. et al. (2014) T*wo different tools for three-dimensional mapping: DE-based scan matching and feature-based loop detection*. In: Robotica. 2014, vol. 32, No.1, pp. 19-41.

Grisetti, G. et al. (2010) *A tutorial on graph-based SLAM*. In: IEEE Intelligent Transportation Systems Magazine. 2010, vol. 2, No. 4, pp. 31-43.

Gutmann, J. S. - Konolige, K. (1999) *Incremental mapping of large cyclic environments*. In: Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99. IEEE, 1999, pp. 318-325.

Kudriashov, A. et al. (2020) *SLAM Techniques Application for Mobile Robot in Rough Terrain*. Springer International Publishing 2020. ISBN 978-3-030-48980-9.

Kümmerle, R. et al. (2009) *On measuring the accuracy of SLAM algorithms*. In: Autonomous Robots. 2009, vol. 27, No.4, pp. 387-407.

Lu, F. - Milios, E. (1997) *Globally consistent range scan alignment for environment mapping*. In: Autonomous robots. 1997, vol. 4, No. 4, pp. 333-349.

Montemerlo, M. - Thrun, S. (2007) *FastSLAM: A scalable method for the simultaneous localization and mapping problem in robotics*. Springer 2007. ISBN 978-3-540-46399-3.

NVIDIA Autonomous Machines (2021) Jetson Nano Developer Kit (2021), [online] Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit (accessed on 30 May 2021).

Prokhorov, D. et al. (2019) *Measuring robustness of Visual SLAM*. In: 2019 16th International Conference on Machine Vision Applications (MVA). IEEE 2019. pp. 1-6.

Raspberry Pi (Trading) Ltd. (2019) Raspberry Pi 4 Model B Datasheet. Available at: https://datasheets.raspberrypi.com/rpi4/raspber-ry-pi-4-datasheet.pdf

Skrzypczyński, P. (2009) *Simultaneous localization and mapping: A feature-based probabilistic approach*. In: International Journal of Applied Mathematics and Computer Science. 2009, vol. 19, No. 4, pp. 575-588.

Shanghai Slamtec CO. (2020) HANGHAI SLAMTEC CO. RPLIDAR A1. Low cost 360 degree laser range scanner. Introduction and datasheet. Model: A1M8. Available at: http://bucket.download.slamtec.com/7fe7e3656e811a-b1a645753af40809f05fa7ddcd/LD108_SLAMTEC_rplidar_datasheet_A1M8_v2.4_en.pdf (accessed on 14 April 2022).

Sensonor AS. (2013) STIM300 Inertia Measurement Unit. Available at: https://www.sensonor.com/media/1132/ts1524r9-datasheet-stim300.pdf, (accessed on 14 April 2022).

Stachniss, C. (2009) *Robotic mapping and exploration*. Springer 2009, vol. 55.

Thrun, S., et al. (2003) *A system for volumetric robotic mapping of abandoned mines*. In: 2003 IEEE International Conference on Robotics and Automation. IEEE, 2003, pp. 4270-4275.

Valencia, R. - Andrade C., J. (2018) Mapping, planning and exploration with Pose SLAM. Berlín: Springer, 2018.