Trajectory Construction for Autonomous Robot Movement based on Sensed Physical Parameters and Video Data

Grigorij Rego, Nikita Bazhenov, Dmitry Korzun
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
{rego, bazhenov, dkorzun}@cs.karelia.ru

***Abstract*—Positioning of a robot in its autonomous movement is applied to the automatic control and trajectory planning. There are two types of systems: sensed data come from physical parameters sensors and video cameras. In this paper, we consider the integration of both data sources, providing a novel way for trajectory construction. The sensed data are collected from the IMU-sensor, camera, and Lidar. We present an architecture for collecting data from many sources and for using the data to evaluate the position of a mobile robot. The results of our early experiments confirm the feasibility of the data integration when a robot moves between special points with different sequences of actions. We also present an algorithm for the movement of a mobile robot between given points without loss of positioning accuracy at long distances. The accuracy is not only affected the Euclidean coordinates, but the orientation angle is also improved. As a result, a robot can be used in narrow indoor, where a special point can be reached only at a certain orientation angle.**

I. Introduction

The problem of controlling the movement of mobile robot is important. One of the biggest challenges facing researchers is the creation of unmanned vehicles. Trajectory control and vehicle navigation are particular tasks of creating unmanned vehicles. Humanity needs not only unmanned vehicles, but also unmanned robots that could autonomously perform some useful operations. It is a global task for researchers who engage robot navigation.

In recent years, many studies have been published on this issue. There are also many classifications of the task. The task can be classified by environment: indoor or outdoor; by type of solution: algorithms and sensors; by the object to which the solution is applied: stationary or mobile; by sensors that are used: portable or stationary. Currently, there is no universal way to solve this problem for all types of applications.

This article discusses the problem of positioning a mobile robot indoors. The task can be described as follows: mobile robot moves indoor between some point. Robot position can be denoted as vector q=[x_c,y_c,θ] where x_c and y_c are coordinates of center of mass and θ is an orientation angle of robot. We have n points in the room and for each of them robot has some restrictions for coordinates. For example, a robot must get to some point and have the right orientation to carry out some operations. To do this, we need to accurately determine the current coordinates of the robot and its orientation.

There is a similar task for tracking a mobile robot. If the trajectory of the robot can be specified with absolute accuracy, then it would be possible to determine the location of the robot at any time. There are many researches of this task. They use stability analysis [1], [2], [3] and neural networks [4]. Trajectory stability analysis is an effective method for problems in which there is a periodically repeating trajectory. However, there are many tasks, including ours, in which the trajectory can be arbitrary. In such tasks, those methods do not achieve the best results. Neural networks require large computing resources. At the same time, mobile robots are usually equipped with cheap single board computers with few computing resources.

For the trajectory construction problem, we consider the integration of both data sources—sensed data come from physical parameters sensors and from video cameras. Such integration enables a novel way for trajectory construction. The sensed data are collected from the IMU-sensor, camera, and Lidar. The proposed architecture supports collecting data from many sources. The collected data are used to evaluate the position of a mobile robot. The results of our early experiments confirm the feasibility of the data integration when a robot moves between special points with different sequences of actions. We also present an algorithm for the movement of a mobile robot between given points without loss of positioning accuracy at long distances. The accuracy is not only affected the Euclidean coordinates, but the orientation angle is also improved. As a result, a robot can be used in narrow indoor, where a special point can be reached only at a certain orientation angle.

The rest of the paper is organized as follows. Section II overviews existing approaches to solving the trajectory construction problem. Section III introduces our architecture that supports trajectory construction in autonomous robot movement. Section IV presents our theoretical algorithm that aims at more effective trajectory construction. We define criteria to evaluate the algorithm and our early experiments that confirm the feasibility. Finally, Section V concludes the paper.

II. Related Work

We analyzed articles by other authors and found several works that are related to ours. In each of them, we found something important for our research and applied it. Currently, many researchers are experimenting with multisensory systems, as they are more effective than single-sensor systems. However, there are studies in which scientists are trying to improve the performance of individual sensors.

The positioning problem has a number of related problems, such as: redundancy of data received from sensors [5]; selection of the most efficient data processing algorithm [6]; choosing a framework for data transfer [7], etc.

The task of mobile robot positioning also has many solutions. In [8] researches used IMU-sensor with Lidar for better positioning of mobile robot. Results of experiments showed that complex system is more efficient for positioning than IMU-sensor or Lidar separately.

In [9] researchers combined Lidar, odometer and IMU-sensor with Kalman filter and used the idea of an external and internal sensors with relative and absolute positioning.

In [10] researchers applied GPS, IMU, and visual odometry. However, their study differs in that they did not work indoors. They worked outdoor.

In [11] researchers also used multisensor approach for mobile robot localization. They collected data from IMU, Ultra-Wide Band (UWB) and Lidar technology. Sometimes it leads to delays, which is unacceptable for the conditions of our task.

In [12] researchers used data from IMU-sensor, optical mouse sensor and wheel encoder and fuse it for input to extended Kalman filter method. This allowed them to get an error of less than 0.6% at a distance of 34 meters. IMU-sensors are designed for accurate positioning, but they have one disadvantage: IMU-sensors accumulate measurement error. At the same time, IMU-sensors have one important advantage. IMU-sensors work with sufficient accuracy at short distances. Unlike sensors, cameras work well over long distances. However, they are not very effective at close range.

Video cameras in robotics often play a navigation role for orienting a mobile robot in space and for recognizing various obstacles on the path of movement. A low-resolution video camera can be used to control a mobile smart robot in a confined space. The authors [13] used images from an RGB-D camera that were transmitted to the robot to detect and avoid obstacles.

Machine vision for a static camera can be used to track the position of the robot [14]. The most important tasks are robot recognition, image for coordinate transformation, coordinates of obstacles and other objects, background recognition and subtraction (sky region, wall region, ground region, etc).

A video camera can be used on a mobile robot as an additional component, for example, monitoring security on the object [15]. In the most advanced version, the camera on the robot can not only record video violations, but also identify the offender by face, for example, in the case when it is an employee working at the facility. Moreover, such solutions can use a cheap video camera and inexpensive components of a mobile robot, which will significantly save costs on security at the facility.

In cases where global positioning is absent (for example, when it is not possible to mount a static video camera), an IMU-sensor or a mobile web camera can be added. There are systems [16] that combine navigation data from sensors and video cameras. Such approach can improve accuracy and efficiency. In particular, data from a video camera should be collected if the distance from the mobile robot to the nearest object is too great, and the characteristics of the sensor or Lidar are not sufficient to achieve this distance.

In [17] researches used new approach for calibration camera. They used mirror and reflection of feature points. This method gives high accuracy, but it can be applied only if we have mirror surface indoor where robot works.

In [18] a new method of calibrating the camera and sensor IMU is presented. The presented results make it possible to reduce the error several times. However, the scope of this method is very limited. You need to have an object directly in front of the camera, to which the camera will be guided.

In [19] this study researches compare approaches for navigation on mobile robots with IMU-sensors. In our work we use them results from section "Odometry and Full IMU with Motion Constraints". They described math model for process of moving mobile robot and show simulation results of them methods. We plan to use some parts of that methods, but this is not enough for our purposes and we want to modify it with camera and LiDAR.

In [20] this study researchers compare different approaches for mobile robot navigation. Difference between of approaches is using the Kinect motion sensor. The data obtained with and without it is compared. In our case, a different set of sensors is used, so the methods of these researchers are not completely suitable for us. However, we apply the general scheme of research and the results of experiments with single sensors and improved algorithms for our task.

In [21] the researchers made a series of experiments to improve vehicle positioning based on MIMU. They used one type of sensor because their goal was to create the most reliable system that would work in almost any environment. In our study, we use similar algorithms: pseudo-acceleration removal procedure and TVU corrections. However, instead of MIMU, we use IMU, so the obtained accuracy is not enough to be limited to these methods.

In [16] researchers used the fusion data from camera and onboard sensors of mobile robot. The researchers concluded that no single sensor can provide sufficient accuracy. We also use this idea in our research. However, this study has one significant disadvantage. All experiments were carried out at very small distances less than one meter. At such distances, the error of the IMU-sensor does not have time to accumulate and this significantly improves the result. Our study requires a mobile robot to move long distances (up to several hundred meters), so this method is not enough for this purpose.

In this study, we consider a comprehensive solution with a new architecture in which the IMU-sensor, Lidar and the camera complement each other. Existing studies are investigating positioning by location coordinates without paying much attention to the orientation coordinate. However, in real applications, when a robotic arm is installed on a mobile robot, it is often necessary that the robot is not only at the desired point, but also at the desired angle of rotation. Such robots are often used at warehouses and sorting points to solve logistics problems.

III. Architecture

In this section we propose an event-driven approach based on the generation of events at the periphery based on data received from various sensors [22]. This approach is the most optimal, given the large amount of information supplied to the board of the mobile robot. Furthermore, this approach allows the use of everyday mobile devices and sensors, for example, web cameras [23].

First we need to define the functional roles involved in the calculations. Such roles of handlers are described in Fig. 1. Entities that are mediators of information are at different levels of processing. The developer is encouraged to select data handlers from the above example and highlight the functional roles that these handlers should perform.

The data publication level consists of data collectors: “collectors”—processors that receive data from the outside (for example, video sensors or sensors), data extractors—software modules that extract data from sensors, nodes - data transmitters between other software objects, gateways - data senders (for example, transmission over the Internet to another communication site). In our example, the data collectors video

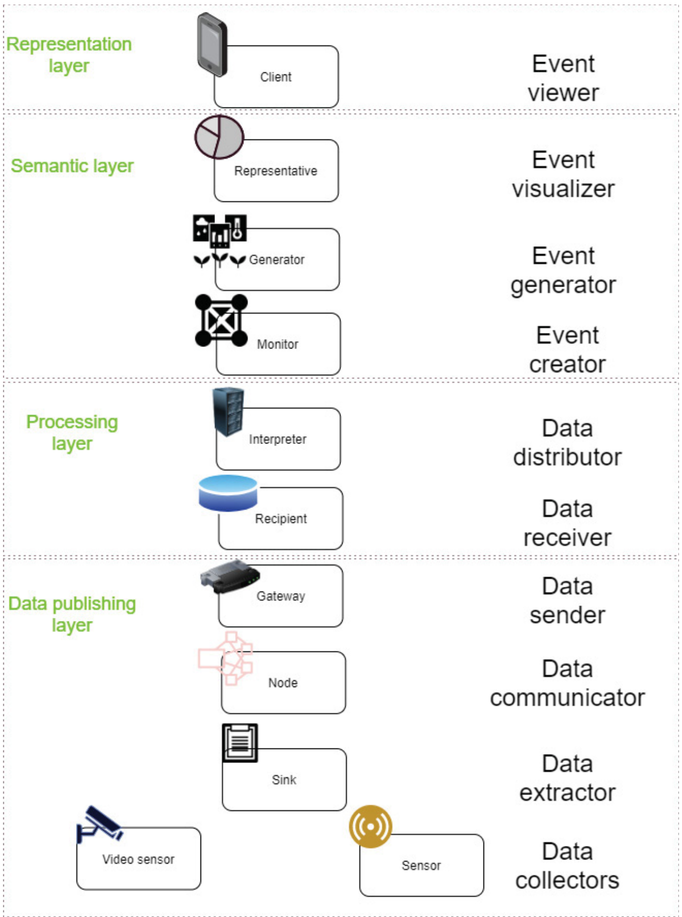

Fig. 1. Layered data architecture

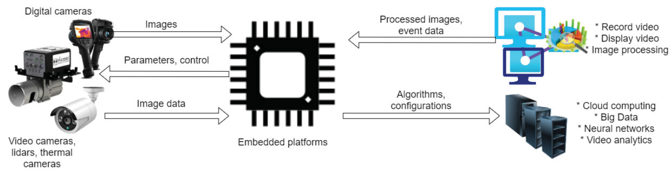

Fig. 2. Edge computing architecture

sensor is represented as a Raspberry Pi video camera, and the sensors are represented as IMU-sensor and LIDAR sensor. In the simplest case, the Raspberry Pi can perform several tasks at once, for example, extract, filter and formalize data coming from heterogeneous sensors. Such operations do not require complex processor calculations and can be implemented on a low-performance computer (even on older versions of Raspberry). As a Gateway, a router that operates at a frequency of 2.4 or 5 GHz can be used.

The processing layer consists of a data sink (it can be a database or storage in the file system, both local and remote) and a data interpreter (a server or broker that analyzes raw data from the database, selecting only the necessary data, for example, for a period or at regular intervals). In our example, Recipient and Interpreter are represented by a mid-range laptop. Basic calculations can be done on this laptop. However, in the case of using algorithms that are complex in terms of computations, an FPGA board or a neural accelerator can be used. This is also recommended in the case of processing data received from 2-3 mobile robots (using more Raspberry Pi). Due to their compactness, neuroprocessors can be attached to the mobile robot itself or connected via USB.

The semantic layer consists of a monitor that monitors incoming (processed by the server) data and converts it into simple events. A generator based on many simple events creates complex events. A representative creates an interface from complex events that are valuable from the user’s point of view. In our example, Monitor is presented on both a Raspberry and a laptop. Simple events that do not require additional processing can be monitored on the on-board computer. In this case, the processing layer (which usually requires complex calculations) can be ”skipped” for some simple tasks. However, complex events that are generated based on basic events usually require the most powerful computing resources. Event visualization can be deployed on a device that has an output to the screen. Usually this is the screen of the same laptop, however, in special cases it can be an on-board Raspberry Pi display.

The presentation layer is made up of clients who view and interact with events. Such a scheme is extensible and can be supplemented with other functional roles, for example, in the case when events consist of several levels of data and there is a need to create more complex analyzing structures. In our example, Client (the web interface) was deployed on a laptop. For the web part, remote servers can be used with the ability to connect globally from the outside (if the system is designed for remote monitoring of events). However, it is not profitable to deploy the web on a Raspberry Pi due to limited computing resources.

In a view of the huge amounts of data and growing comput-

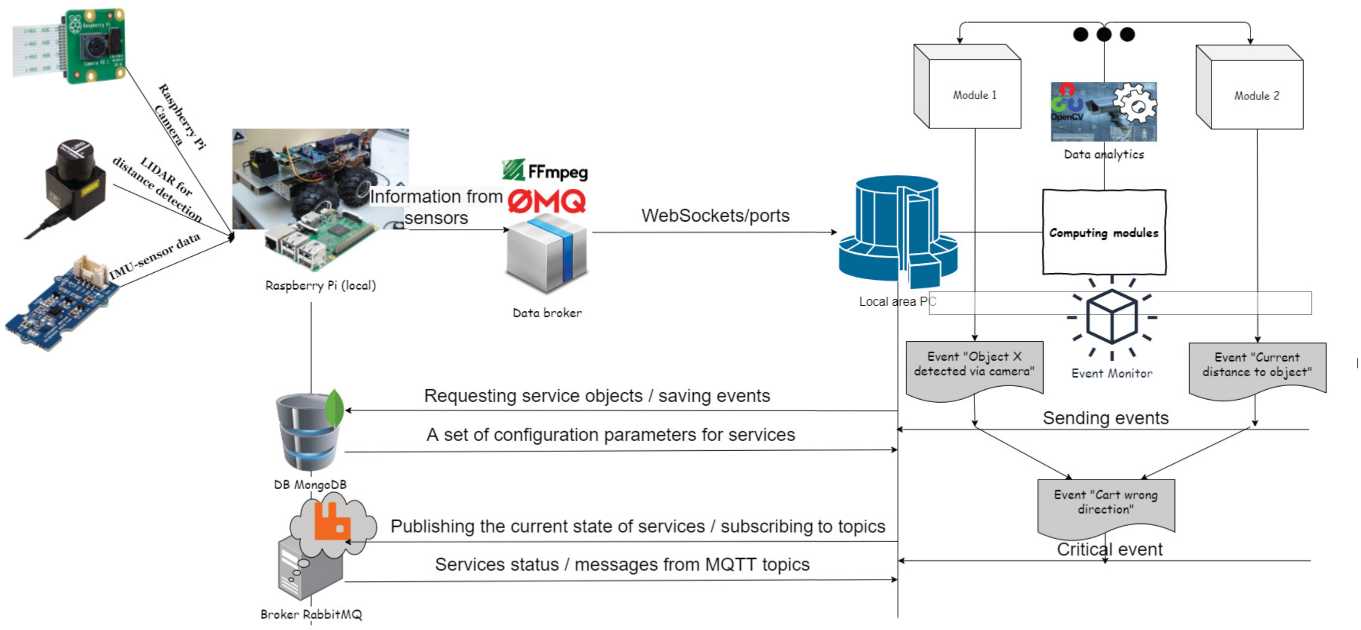

Fig. 3. The architecture of data processing on Raspberry Pi

ing power [24], it is necessary to imagine how the considered data layers will be processed and stored on various computing devices. Examples of configurations for such devices are as follows.

- Edge computing: Sensor (= Video Sensor) - LAN (Local Area Network) - all basic calculations are performed on end devices, that is, "next to" the sensors.
- Fog computing: Sensor - LAN - DPC (Data Processing Center) - calculations are performed using a decentralized system (data center) that processes data "near" the sensors.
- Cloud computing: Sensor - LAN - DPC - CP (Cloud Platform) - calculations are performed in the cloud, on the cloud platform, "far" from the sensors.

Video processing and image analysis are computationally intensive. Small microcomputers like the Raspberry Pi or Arduino do not have enough computing power to read and process large amounts of sensor data. One of the solutions is to transfer video analytics computations to the data processing center (Fig. 2), but this algorithm has a large delay in the mobile robot's response to obstacles. In our solution based on Edge-devices we use the local neuro-accelerator Google Coral to reduce the load on the Raspberry Pi.

The architecture of data processing on Raspberry Pi is shown in Fig. 3. Raspberry Pi camera, Lidar and IMU-sensor send data to the Raspberry Pi microcomputer (which is installed on the mobile robot). It can recognize patterns and calculate distance to the near objects in the current working area. Raspberry Pi then sends data using the ZeroMQ broker that generates video streams on the ports and transmits it to a more powerful computing device - the local PC. Such PC implements all the necessary analytics. Depending on the states, the event monitor, based on the processed data, generates events that an object was detected or current distance to object and sends it back to Raspberry Pi and in the MongoDB database for storage and subsequent retrieval as needed, and to the RabbitMQ message broker to receive notifications on the browser. Depending on the current statuses, a regular event or a critical one will be sent. A critical event happened if the cart going on the wrong direction or if insurmountable obstacle was found.

IV. Experiments

A. Mathematical model

Let us describe the original problem and the upcoming experiments. We are faced with the task of testing an algorithm that allows positioning the robot with high accuracy.

We have some kind of rectangular indoor in which there are no thresholds. Also in this indoor it is always possible to move in a straight line, there are no curvilinear trajectories. An example of such a indoor is a warehouse with rows of shelving. The mobile robot only moves between predetermined points. Let's denote it as special points. For each special point (SP), the x, y and ?-rotation coordinates are known, with which the robot should be at the point. For each special point i, deviations of $?_{ix}$, $?_{iy}$ and $?_{i\theta}$ are allowed. The deviations could be either the same or different for each special point. It depends on the conditions of the specific problem for which the algorithm is applied.

At the initial moment of time, the mobile robot is at its parking lot. For definiteness, let's denote this point as SP_0 with coordinates x=0, y=0, ?=0. The input to the robot is a sequence of special points that the robot must visit. It looks like array C of coordinates where variables are numbers. For example, x_1, y_1, θ_1. Another array E contains variables of allowed deviation. For example, $?_{1x}$, $?_{1y}$, $?_{1\theta}$. When the robot arrives at a special point, it compares its current coordinates x_{cur}, y_{cur}, θ_{cur} with the specified coordinate range $x_i \pm ?_{ix}$, $y_i \pm ?_{iy}$, $\theta_i \pm ?_{i\theta}$, where i is current special point. If the current coordinates are within the specified range, then the robot continues to move in accordance with the algorithm, otherwise it corrects its position.

The robot knows the coordinates of all the special points, based on this information, the robot builds its path. For definiteness, we will assume that the robot can move forward or turn one of the angles: 90 degrees to the right, 180 degrees, 90 degrees to the left. We will also assume that the robot spends 10 seconds at each particular point on its path, after which it continues to move. This is necessary in order to compare the absolute values of coordinates that are known to researchers with the relative ones that the robot has.

B. Path Control and Positioning Algorithm

The main sensor, the readings from which will be the main ones when orienting at a particular point, is the IMU sensor. The main problem of such sensors is the accumulation of errors, however, using the data of the camera and rangefinder, one can reset the accumulated error. At short distances (less than one meter), we plan to use the extended Kalman filter. Over long distances, calibration will be carried out using a camera and Lidar.

The URG-04LX-UG01 laser rangefinder allows one to determine the distance to objects with a range of up to 4 meters. The measurement error is 3 percent of the distance when the distance to the object is more than 1 meter and 0.03 meters when the distance is less than one meter. The rangefinder allows you to measure distances to objects at a speed of 10 measurements per second. Rangefinder data can help at medium distances, when a feature point falls into the camera's dead zone when approaching it.

Before starting the movement, after receiving a sequence of points, the algorithm calculates an approximate route of movement. The route contains a sequence of movements and turns. Each speed corresponds to the length of time during which the mobile robot must move. Each turn corresponds to the direction of rotation and the number of degrees that the mobile robot must turn. Also, each action contains a motion stage parameter. For example, moving from the base to the first singular point is the first stage, from the first singular point to the second is the second, and so on.

The main idea of the algorithm is as follows. We have some marks that can be recognized by the camera image. The marks are located on the walls at a certain height and distance from the point on the floor that the mobile robot should occupy. If the image from the camera contains the image of the mark, then at large distances (more than 4 meters) it is possible to analyze the video sequence and, on its basis, determine the distance to the special point. If the mark is not visible on the camera image, then the mobile robot moves along a pre-calculated route, in accordance with the stage at which the robot's route is.

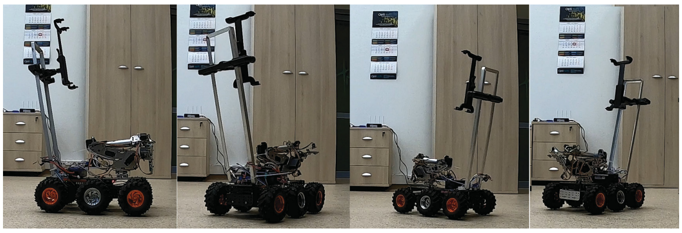

Fig. 4. The movement of a mobile robot around its axis

When the robot approaches a special point at a distance of about 10-20 centimeters, it is necessary to calibrate the IMU sensor. We need to define the parameters for position and orientation. In order to determine the orientation, one can take an image from the camera and analyze the position of the robot relative to the mark. When the robot is directly in front of the mark, the image must be exactly rectangular. If the mobile robot is rotated with respect to the mark by a non-zero angle, then the image of the mark will be curved. In addition, one can get data on the distance to different ends of the mark from the Lidar. By filtering camera and Lidar data with extended Kalman filter, one can calibrate the IMU sensor by orientation. Coordinate parameters are calibrated using the same mechanism.

We used Raspberry Pi Camera v2 that allows to read images with a Sony IMX 219 PQ sensor type, a resolution of 1280x720 (sufficient for our experiment) and a light weight of 3 grams. We used camera at an average distance (from 4 meters to 10 centimeters), a laser rangefinder becomes the main sensor, which corrects the trolley on the approach to a special point.

At the final stage of movement, the IMU-10dof becomes the main sensor. From its work depends the correct position of the mobile robot more than from other sensors, therefore its accurate calibration is very important.

More formally, the algorithm can be described as follows:

1) Turning on the robot.
2) Built-in sensor calibration.
3) The robot receives a sequence of special points.
4) If the robot sees the next special point, then the path is built based on the indicators of the sensors
5) Otherwise, the robot does not see the next special point, the path is built on the basis of odometry.
6) The movement begins to the next special point
7) If at step 4 the robot did not see the special point, then it checks every 3 seconds if the special point has appeared in the visibility zone. If it appears, step 4.
8) At the entrance, the robot calibrates the data from the sensors and checks the coordinates. If the coordinates are in the allowable range, then the next special point is sent to the robot, Step 4.
9) Otherwise Step 4 with the current special point.
10) If the path is complete, robot returns to base.

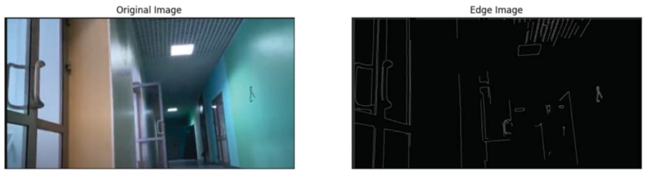

Fig. 5. Detecting walls and objects via camera on a mobile robot

During the operation of the algorithm, all data is transmitted and processed as described in the Architecture section.

C. Early and Future Experiments

Let us describe the conditions for applying any algorithm, as well as the practical needs in such an algorithm. In the early experiments, we studied a mobile robot on which a manipulator was installed (Fig. 4). The experiment consisted in making the cart rotate 180 degrees around its axis. Since only the base link was rigidly fixed to the manipulator, the inertia generated by the manipulator was different all the time, since the rest of the links began to move.

This problem can be solved by holding a certain position of the links with the help of servomotors. However, this leads to a huge increase in battery consumption as well as overheating of the servos. Because of this, in practice, such a solution is not feasible, at least for cheap robots.

The second possible solution to the problem of changing inertia from the movement of the manipulator links is the development of a mathematical model of inertia change using dynamic manipulation methods. If it is possible to build a model of inertia change, then it will be possible to find a control action that will compensate for the resulting inertia. A similar solution is planned to be investigated in the future.

In particular, at the first steps of the experiment, using machine vision technologies and OpenCV algorithms, we tried to recognize objects (doors, walls) that arise during the movement of the cart in the corridor of an office building shown in Fig. 5. The approximate recognition algorithm was:

- Image translation to GrayScale with Canny border detector, noise smoothing;
- Calibration of the camera (some of it require measurement of objects in ideal conditions: doors, door handles, scale ratio);
- Detection of boundaries of space in real time, building a diagram of the space (walls / doors / ceiling);
- Determining the distance to the wall;
- Determination of the trajectory of movement (where we can or cannot go).

Another problem found in our early experiments is the friction of the wheels of the mobile robot against the floor surface. A mobile robot, especially if it contains additional equipment, has a minimum value of the force that must be applied for the robot to start moving. However, if the friction with the surface is large, then the minimum force becomes too large to accurately control the robot. This leads to the fact that it is impossible to predict the behavior of the robot in a particular situation.

In the first version of the research, we used IMU-10DOF and 2 cameras: one was external and static, the second was fixed on a mobile robot and moved with it. During the experiments, we tried to achieve two goals: 1) turn the mobile robot with high accuracy; 2) correct the movement of the mobile robot so that it moves in a straight line.

The rotation of the mobile robot could not be standardized due to the reasons described above. However, we tried to use another mobile robot with the most robust design and a fixed Raspberry Pi camera on board.

To achieve the second goal, we conducted a series of experiments, during which two types of tuning were performed: mechanical and software. Mechanical adjustment included changing the position of the camera and battery. This made it possible to change the center of mass of the mobile robot and smooth out the resulting inertia.

In addition, we received data from the IMU sensor and camera. As a result of the analysis of the data obtained, using the Kalman filter, it was possible to achieve a deflection of the mobile robot by no more than 3 degrees with a rectilinear movement at a distance of 10 meters.

At the next stage of experiments, we plan to modify several technological aspects: modification of a mobile robot (the most efficient arrangement of components - a battery, a camera, a microcomputer), an increase in the performance of the Raspberry Pi - since the computational capabilities of a microcomputer processor are not enough for object recognition, it is planned to integrate part of the calculations on a portable neuro accelerator (for example, Google Coral); improvement of the image segmentation algorithm (the best separation of the wall and the floor), the best noise filtering algorithms for the IMU-sensor.

Based on our early experiments, we put forward the following hypotheses.

1) If the deviation from the required coordinates is small (within a few tens of centimeters), then the mobile robot will not be able to make the necessary movement for correction. It is necessary to drive a couple of meters back/forward and repeat the algorithm again.
2) The given set of sensors is capable of effectively solving problems of autonomous movement.
3) The given set of sensors is capable of solving problems in rooms where there are dynamic obstacles.

Corresponding experiments are also planned:

1) Comparison of attempts to correct the position of the robot at short distances locally and using a new run of the algorithm.
2) A series of runs of the algorithm for a different set of singular points and input deviation parameters to find out the accuracy of the solution.
3) The same as in 2, but on condition that there are people in the room.

V. Conclusion

This paper considered the problem of trajectory construction based on positioning a mobile robot. We considered the tasks of navigating a mobile robot when moving between special points. Our early experiments used a camera built into a mobile robot, an inertial sensor and a Lidar. Our goal is that the sensed data are used in integration. At long distances (over 4 meters) the camera works best, at medium distances (from 10 centimeters to 4 meters) the Lidar works best, at short distances it is necessary to use an IMU sensor.

Our primary contribution is the proposed architecture for trajectory construction from the integrated sensed data. Our early experiments to evaluate the architecture were performed without a lidar. The results show the possibility to set up the linear motion of a mobile robot in various conditions.

We formulated further requirements for the design of mobile robots. In particular, robots should not have a large mass and friction of the wheel against the surface at the same time. In addition, the equipment that is on the robot must either be stationary or not affect the center of mass of the mobile robot. Data received from IMU, camera, and Lidar. Our plans are to carry out experiments with sensed data from all three sensors, processing them with an extended Kalman filter, and calibrating the values relative to each other.

Acknowledgment

This development was implemented in Petrozavodsk State University (PetrSU) with financial support of the Ministry of Science and Higher Education of Russia within Agreement no. 075-11-2019-088 of 20.12.2019 on the topic "Creating the high-tech production of mobile microprocessor computing modules based on SiP and PoP technology for smart data collection, mining, and interaction with surrounding sources". This research study is supported by RFBR (research project # 19-07-01027) in part of edge-centric computing models for data processing. The work is in collaboration with the Artificial Intelligence Center of PetrSU.

References

[1] R. Anushree and B. S. Prasad, "Design and development of novel control strategy for trajectory tracking of mobile robot: Featured with tracking error minimization," in *2016 IEEE Annual India Conference (INDICON)*, 2016, pp. 1–6.

[2] A. Y. Lee, G. Jang, and Y. Choi, "Infinitely differentiable and continuous trajectory planning for mobile robot control," in *2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2013, pp. 357–361.

[3] X.-Z. Jin, J.-Z. Yu, L. Zhou, and Y.-Y. Zheng, "Robust adaptive trajectory tracking control of mobile robots with actuator faults," in *2019 Chinese Control And Decision Conference (CCDC)*, 2019, pp. 2691–2695.

[4] M. Asai, G. Chen, and I. Takami, "Neural network trajectory tracking of tracked mobile robot," in *2019 16th International Multi-Conference on Systems, Signals Devices (SSD)*, 2019, pp. 225–230.

[5] K. Krinkin and A. Filatov, "Correlation filter of 2d laser scans for indoor environment," *Robotics and Autonomous Systems*, vol. 142, p. 103809, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889021000944

[6] A. Huletski, D. Kartashov, and K. Krinkin, "Vinyslam: An indoor slam method for low-cost platforms based on the transferable belief model," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 6770–6776.

[7] A. Gavrilov, M. Bergaliyev, S. Tinyakov, and K. Krinkin, "Analysis of robotic platforms: Data transfer performance evaluation," pp. 437–443, 2021.

[8] D. R.-Y. Phang, W.-K. Lee, N. Matsuhira, and P. Michail, "Enhanced mobile robot localization with lidar and imu sensor," in *2019 IEEE International Meeting for Future of Electron Devices, Kansai (IMFEDK)*, 2019, pp. 71–72.

[9] Z. Li, Z. Su, and T. Yang, "Design of intelligent mobile robot positioning algorithm based on imu/odometer/lidar," in *2019 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, 2019, pp. 627–631.

[10] G.-S. Cai, H.-Y. Lin, and S.-F. Kao, "Mobile robot localization using gps, imu and visual odometry," pp. 1–6, 2019.

[11] Y. Gao, F. Wang, J. Li, and Y. Liu, "Localization of mobile robot based on multi-sensor fusion," pp. 4367–4372, 2020.

[12] Y. Liu, Y. Ou, and W. Han, "Mobile robot localization based on optical sensor," in *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2019, pp. 874–879.

[13] S. Kebir, H. Kheddar, M. Maazouz, S. Mekaoui, A. Ferrah, and R. Mazari, "Smart robot navigation using rgb-d camera," in *2018 International Conference on Applied Smart Systems (ICASS)*, 2018, pp. 1–6.

[14] R. J. C. Saclolo, J. B. Delos Reyes, N. N. R. B. Formeloza, J. I. O. Abaca, K. K. Serrano, A. Dela Cruz, E. Roxas, and R. R. Vicerra, "Machine vision system, robot hardware design and fuzzy controller design for autonomous multi-agent mobile robot platform," in *2015 International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, 2015, pp. 1–6.

[15] F. Khalid, I. H. Albab, D. Roy, A. P. Asif, and K. Shikder, "Night patrolling robot," in *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 2021, pp. 377–382.

[16] A. Novitsky and D. Yukhimets, "The navigation method of wheeled mobile robot based on data fusion obtained from onboard sensors and camera," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, 2015, pp. 574–579.

[17] G. Panahandeh and M. Jansson, "Imu-camera self-calibration using planar mirror reflection," in *2011 International Conference on Indoor Positioning and Indoor Navigation*, 2011, pp. 1–7.

[18] X. Ouyang, Y. Shi, You, and Zhao, "Extrinsic parameter calibration method for a visual/inertial integrated system with a predefined mechanical interface," *Sensors*, vol. 19, p. 3086, 07 2019.

[19] T. Fauser, S. Bruder, and A. El-Osery, "A comparison of inertial-based navigation algorithms for a low-cost indoor mobile robot," in *2017 12th International Conference on Computer Science and Education (ICCSE)*, 2017, pp. 101–106.

[20] H. G. Y. X. Mingjing Gao, Min Yu, "Mobile robot indoor positioning based on a combination of visual and inertial sensors," *Information Systems Frontiers*, vol. 19, Apr. 2019.

[21] A. Mikov, A. Moschevikin, and R. Voronov, "Vehicle dead-reckoning autonomous algorithm based on turn velocity updates in kalman filter," in *2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, 2020, pp. 1–5.

[22] N. Bazhenov and D. Korzun, "Event-driven video services for monitoring in edge-centric internet of things environments," in *Proc. 25th Conf. Open Innovations Association FRUCT*, Nov. 2019, pp. 47–56.

[23] N. A. Bazhenov and D. G. Korzun, "Use of everyday mobile video cameras in IoT applications," in *Proc. 22nd Conf. Open Innovations Association FRUCT*, May 2018, pp. 344–350.

[24] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.