Nexo Revista Científica

Vol. 33, No. 02 pp. 651-671/Diciembre 2020

ISSN-E 1995-9516
Universidad Nacional de Ingeniería

http://revistas.uni.edu.ni/index.php/Nexo
https://doi.org/10.5377/nexo.v33i02.10800

Motion Planning Algorithm for a Mobile Robot with a Smart Machine Vision System

Algoritmo de planificación de movimiento para un robot móvil con un sistema de visión artificial inteligente

Alexey I. Martyshkin

Candidate of Technical Sciences, docent, Associate Professor of sub-department «Computers and systems», *Penza state technological University , (440039, Russia, Penza, Baydukov Proyezd / Gagarin Street, 1a/11, e-mail:* alexey314@yandex.ru

**Corresponding author email: alexey314@yandex.ru*

(recibido/received: 16-July-2020; aceptado/accepted: 19-September-2020)

ABSTRACT

This study is devoted to the challenges of motion planning for mobile robots with smart machine vision systems. Motion planning for mobile robots in the environment with obstacles is a problem to deal with when creating robots suitable for operation in real-world conditions. The solutions found today are predominantly private, and are highly specialized, which prevents judging of how successful they are in solving the problem of effective motion planning. Solutions with a narrow application field already exist and are being already developed for a long time, however, no major breakthrough has been observed yet. Only a systematic improvement in the characteristics of such systems can be noted. The purpose of this study: develop and investigate a motion planning algorithm for a mobile robot with a smart machine vision system. The research subject for this article is a motion planning algorithm for a mobile robot with a smart machine vision system. This study provides a review of domestic and foreign mobile robots that solve the motion planning problem in a known environment with unknown obstacles. The following navigation methods are considered for mobile robots: local, global, individual. In the course of work and research, a mobile robot prototype has been built, capable of recognizing obstacles of regular geometric shapes, as well as plan and correct the movement path. Environment objects are identified and classified as obstacles by means of digital image processing methods and algorithms. Distance to the obstacle and relative angle are calculated by photogrammetry methods, image quality is improved by linear contrast enhancement and optimal linear filtering using the Wiener-Hopf equation. Virtual tools, related to mobile robot motion algorithm testing, have been reviewed, which led us to selecting Webots software package for prototype testing. Testing results allowed us to make the following conclusions. The mobile robot has successfully identified the obstacle, planned a path in accordance with the obstacle avoidance algorithm, and continued moving to the destination. Conclusions have been drawn regarding the concluded research.
Keywords: mobile robot, smart system, machine vision, planning algorithm, prototype, robotics, decision theory.

RESUMEN

Este estudio está dedicado a los desafíos de la planificación del movimiento para robots móviles con sistemas inteligentes de visión artificial. La planificación del movimiento para robots móviles en un entorno con obstáculos es un problema con el que lidiar al crear robots adecuados para operar en condiciones del mundo real. Las soluciones que se encuentran en la actualidad son predominantemente privadas y altamente especializadas, lo que impide juzgar qué tan exitosas son para resolver el problema de la planificación eficaz del movimiento. Ya existen soluciones con un campo de aplicación estrecho y ya se están desarrollando durante mucho tiempo, sin embargo, aún no se han observado avances importantes. Solo se puede observar una mejora sistemática en las características de tales sistemas. El propósito de este estudio: desarrollar e investigar un algoritmo de planificación de movimiento para un robot móvil con un sistema de visión artificial inteligente. El tema de investigación de este artículo es un algoritmo de planificación de movimiento para un robot móvil con un sistema de visión artificial inteligente. Este estudio proporciona una revisión de robots móviles nacionales y extranjeros que resuelven el problema de planificación de movimiento en un entorno conocido con obstáculos desconocidos. Se consideran los siguientes métodos de navegación para robots móviles: local, global, individual. En el transcurso del trabajo e investigación se ha construido un prototipo de robot móvil, capaz de reconocer obstáculos de formas geométricas regulares, así como planificar y corregir la trayectoria del movimiento. Los objetos del entorno se identifican y clasifican como obstáculos mediante métodos y algoritmos de procesamiento de imágenes digitales. La distancia al obstáculo y el ángulo relativo se calculan mediante métodos de fotogrametría, la calidad de la imagen se mejora mediante la mejora del contraste lineal y el filtrado lineal óptimo utilizando la ecuación de Wiener-Hopf. Se han revisado las herramientas virtuales, relacionadas con las pruebas de algoritmos de movimiento de robots móviles, lo que nos llevó a seleccionar el paquete de software Webots para las pruebas de prototipos. Los resultados de las pruebas nos permitieron sacar las siguientes conclusiones. El robot móvil identificó con éxito el obstáculo, planificó una ruta de acuerdo con el algoritmo de evitación de obstáculos y continuó avanzando hacia el destino. Se han extraído conclusiones con respecto a la investigación concluida.

Palabras clave: robot móvil, sistema inteligente, visión artificial, algoritmo de planificación, prototipo, robótica, teoría de la decisión.

1. INTRODUCTION

Motion planning for a mobile robot (MR) in the presence of obstacles is a rather urgent problem related to creation of MRs suitable for operation in real-world conditions. To date, the existing solutions are predominantly private, and are highly specialized, preventing us from judging how successful they are in solving the problem of effective motion planning. Certain solutions with a narrow application field are already being developed for a long time, but nevertheless, no major breakthrough has been observed in this domain. Only a systematic improvement in the characteristics of such systems can be noted.

2. THEORETICAL PART

This study is mostly investigative. References (Zavlangas et al., 2000; Chi & Lee, 2011; Kristensen et al., 2001; Ulas & Temeltas, 2013; Voitovich, 2009), (Dergachev, 2017; Kuchersky, 2012; Montemerlo & FastSLAM, 2007; Visilter, 2007; Lokhin et al., 2008) and (Romanov & Gartseev, 2008; Castellanos et al., 2004; Durrant-Whyte, 2002; Montemerlo, 2003; Nist´er et al., 2006; Devyaterikov & Mikhailov, 2015) have been analysed during the study of the subject field with the aim of finding unaffected and unsolved issues. Some issues related to motion planning algorithms for MRs with smart machine vision systems are not adequately reflected in the publications, but certain issues are partially considered in (Gerasimov & Mikhailov, 2012;

Gerasimov, 2015; Gerasimov, 2014; Thrun, 2004; Calonder, 2006).

The main function of an autonomous MR is navigation (motion planning in the field). The solutions used today in household multi-functional robots are expensive, which prevents their mainstream use. The purpose of this paper is to develop and investigate a motion planning algorithm for MR with a smart machine vision system. This study tries to solve this problem by creating a simple MR prototype controlled by a mobile device (MD). And with the rapid development of the MD production line, manufacturers seek to equip these MDs with simple but modern sensors that can be programmed thanks to availability of open sources operating systems. Thus, MD includes several important MR subsystems: environmental data collection subsystem, environmental data analysis subsystem, motion planning subsystem.

Such combination of MD features simplifies the challenge of creating an MR, affordable for a wide range of users, which consists of only two units: MD and the system's executive part. Any wheelbase, which includes a battery, propulsion system, as well as a controller with a transceiver coordinating the system's executive part and receiving control signals, can be used as the executive part.

These issues are relevant today due to the active introduction of various mobile robot design solutions in various spheres and sectors of the economy. Robots with smart machine vision systems are actively used in the Russian Federation and world markets.

The following particular objectives shall be completed to achieve this goal:

1. Review the solutions existing in the global and domestic mobile robotics experience.
2. Review the existing motion planning systems for mobile robots.
3. Create a motion planning system prototype for a mobile robot with a smart machine vision system.
4. Perform prototype testing.
5. Draw conclusions on the results.

Initial data and assets for the research: a three-wheeled Lego Mindstorms NXT platform with two servo motors, Android mobile device with a gyroscope, accelerometer, digital video camera, and Bluetooth module.

3. OVERVIEW OF EXISTING SOLUTIONS

This paper discusses MRs related to household and special purpose groups (Bradski, 1998). During the study of motion planning problem solutions, MR navigation has been found to be a fundamental component, therefore, given the wide range of applications for navigation problem solutions, three types of navigation methods can be distinguished:

1) Global — representing the coordinates of a moving object in a chosen coordinate system, usually an absolute coordinate system.

2) Local — representing the coordinates of a moving object in a local coordinate system, tied to the selected starting position on the path. Widely used in the design of unmanned aerial vehicles.

3) Individual — representing the coordinates of robot's units and assemblies relative to the base unit, or relative to objects located in the robot's operating area (Bradski & Kaehler, 2008).

Taking into account specific aspects of navigation of MRs of various sizes, a relationship exists between the robot size and navigation method used by the robot. For instance, large-sized robots navigate using global navigation methods, and small-sized ones — using individual navigation. Certain solutions exist combining several navigation methods in a single robot. Such method combinations improve the positioning accuracy of a robot as an object. A more detailed study of each navigation method allowed us to find common aspects that can be summarized into a separate feature, which includes a passive navigation system constantly receiving data from external systems both about its own coordinates and about the quantitative properties of movement, and an active navigation system, which determines the position by analysing data from MR sensors and deciding on the further direction and movement speed to complete the current task. An active navigation system is constantly used by robots that use individual navigation methods, and half of the time — by the robots using local navigation methods. Therefore, it can be concluded that MRs using local navigation methods perform a wider range of tasks.

The first models of industrial robots created in the 1960s moved in a strictly designated path along electric cables laid under the floor of a factory building (Bradski, 1998). Creation and introduction of the first machine vision systems allowed to abandon the cables and switch to navigation along bright lines painted on the floor. Using a camera, the robot traced such lines and moved along it autonomously. Other similar concepts have been tested as well (Fukunaga & Hostetler, 1975).

A large number of well-known robotic platforms for MR motion research are available on the market, and we would like to highlight the most interesting projects, such as Lego Mindstorms, iRobot, Thymio. Today, more and more interest is caused by Android-based MDs. The role of MDs in such systems is ambiguous. For example, Meet ROMO project uses MD for user interaction, while environmental data collection, navigation, and motion planning are performed using sensors and a microcomputer built into the platform. In the rest of the listed projects, MDs perform more functions — from data collection to decision making. But, despite this, such MRs still have a rather complex and expensive design. Currently, the actively developed international projects in the household segment of autonomous MRs are Cellbots, IOIO, Meet ROMO, Wheelphone, Botiful. There are virtually no MR analogues utilizing Android mobile devices built in the Russian Federation as of today. The general functional design of MR with MD control is presented in Figure 1.

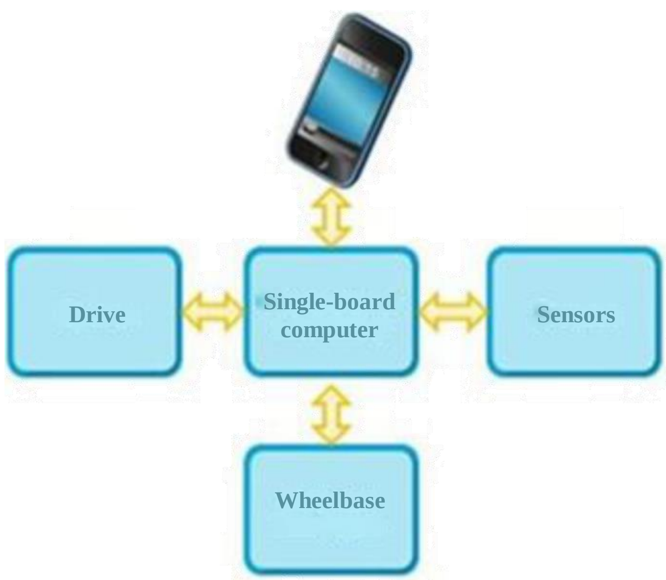

Figure 1: Mobile robot functional design

A single-board computer controls servo motors installed on a wheelbase via digital signals. Sensors detect the feedback needed to control movement. MD acts as a high-level control device

in this system, and in some cases plays the role of a single-board computer and sensors, thereby optimizing the robot's functional design ("Single-board computer" and "Sensors" blocks can be removed, respectively).
During the study of existing MR solutions with MD controls, the average specifications of such devices have been derived, which are shown in Table 1.

Table 1: Specifications of devices

Component	Unit of measurement	Min value	Max value
CPU	Clock frequency (GHz)	1.0	8·1,4
RAM	Memory size (GB)	0.5	2
GPU	Performance (DMIPS/MHz)	1	3
Video camera	Sensor resolution (MP)	5	20

4. MOBILE ROBOT DESIGN

Development of the MR motion planning system pays much attention to the control system design, obstacle recognition and avoidance algorithms. Therefore, the standard solution from LEGO was chosen as the wheelbase and drives, namely, the NXT platform, which allows controlling up to three servo motors and provides easy-to-use sensors, facilitating the creation of MRs of virtually any design. The built-in Bluetooth module provides wireless connection with MR, which greatly simplifies the design. The fundamental concept in MR design for solving the motion planning problem consists in dividing the device into 2 layers: low-level and high-level. Schematic division into low-level and high-level layers is shown in Figure 2.

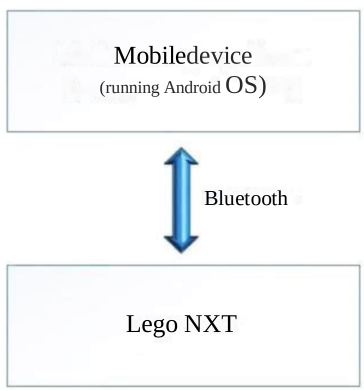

Figure 2: Logical representation levels

The low-level layer consisting of Lego NXT allows controlling the sensors and actuators included in the standard kit (Göbel et al., 2011). In addition, this level is characterized by identical servo motors with high rotation angle positioning accuracy and lejOS OS control. The high-level layer performs calculation of control signals when planning complex system behaviour. Using MD with Android OS allows using all the advantages of this system, determined by the openness of this OS. The connection between the two layers can be implemented via USB and Bluetooth, since both interfaces are fully supported by the Lego NXT platform and any modern MD with specifications similar to those given in Table 1. In order to simplify MR design, we have chosen the Bluetooth interface. Devices of each layer have independent power supply, as the Android device is powered by its own battery, and the Lego NXT control unit can be powered from to options: 6 AA cells or a rechargeable battery). Both devices have optimized power consumption. If the robot is meant to be used for industrial purposes, an additional power source is recommended to be introduced into the circuit, which makes it possible to compensate for the difference in working time of MD and the Lego NXT platform. The Lego NXT controller contains 3 output ports, 4 input ports for sensors, a USB port for downloading and uploading data.

Samsung Galaxy S4 phone has been selected as MD installed into MR. A subject-oriented model describing the position of components in a system is presented in the form of an MR component diagram in Figure 3.

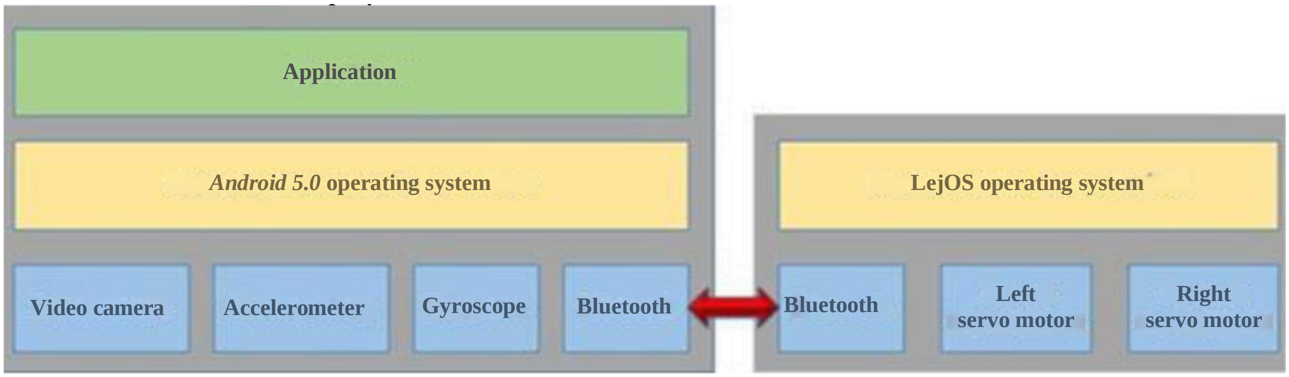

Figure 3: Logical representation diagram for MR components

To simplify the design when solving a research problem, a three-wheeled platform with two independent driving wheels has been assembled from the standard Lego NXT components. Despite the fact that a three-wheeled platform has less stability than a four-wheeled platform, this solution is sufficient for the research task.

When developing a mathematical model of MR with two DC motors rotating the robot wheels, the effects of inertia of the third castor wheel on the robot dynamics were neglected, and drive wheels were assumed as having no slipping motion. Therefore, taking into account the neglects, the third wheel is represented in calculations as an absolutely smooth support at a certain location. Figure 4 shows a diagram of a three-wheeled MR, where L, R are the left and right independent drives, respectively, N is the castor wheel, ω_1, ω_2 are the angular velocities of the left and right wheels, ω_P is the platform's angular velocity relative to the vertical axis. Lego NXT L and R platform servo motors drive the driving wheels.

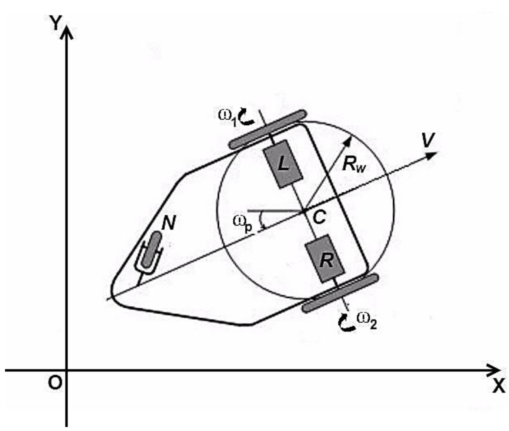

Figure 4: Mobile robot kinematic diagram

In the process of deriving the equations of MR motion MR, its motion will be described by a change in the position of point C. The coordinate system shown in Figure 10 is fixed, and is introduced into the system to solve the robot positioning problem. When moving in a real environment, east direction is taken for the x-axis, and north direction — for the y-axis (Plotnikov, 2010). MR speed is determined according to the following equation:

$$V = \frac{\omega_1 + \omega_2}{2} \cdot r_k, \tag{1}$$

where: ω_1, ω_2 are angular velocities of the left and right driving wheels; r_k is the wheel radius.
At turns and movement angle changes, the angular velocity becomes non-zero, and takes on a value based on angular velocities of the wheels:

$$\omega_p = \frac{\omega_1 - \omega_2}{2} \cdot \frac{r_k}{R_\omega}, \tag{2}$$

where: R_ω is half the platform (wheelbase) width.

Control signals are supplied from the Lego NXT controller to servo motors, therefore, with the known values of servo motor transfer function parameters in terms of velocity W_L and W_R, the ratio of wheel angular velocities to the control voltages can be represented in the form of the following equations using Laplace complex variables:

$$\begin{gathered} \omega_1 = W_L(s)U_1(s) \\ \omega_2 = W_R(s)U_2(s), \end{gathered} \tag{3}$$

where: $U_1(s)$, $U_2(s)$ are Laplace complex variables for input voltage signals supplied to the left and right motors, respectively (Plotnikov, 2010).
Combining the equations obtained in the previous steps, we obtain the operator and structural diagram of the MR's executive part, which is shown in Figure 5.

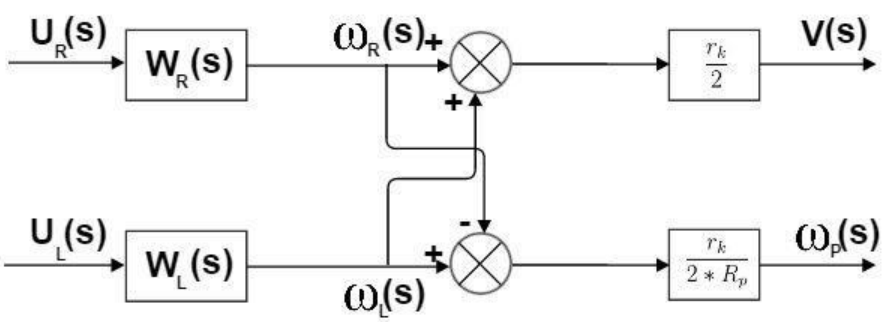

Figure 5: Operator and structural diagram of the system's executive part

Input voltage signals $U_1(s)$, $U_2(s)$, passing through the dynamic elements W_L and W_R, are converted into angular velocities of the wheels ω_1, ω_2, and, passing through the proportional elements, are converted into output variables V(s) and $\omega_P(s)$.
Android SDK 5.0 API level 21 was used to create the MD side application. Android OS represents a component model of processes in the system. When calling an external API application, the data

provider accesses the LINUX kernel via libraries, which provides raw data that is already processed at the application level by means of a requested physical device driver (Strom et al., 2009). All modules and libraries included at the compilation stage are compiled into native C/C++ code before starting.

On the LegoNXT platform side, the control unit is running LejOS OS, which has the following features:

1) Supports .NET, Java, Python, C/C++ programming languages.
2) Library of Java classes for implementing custom application interface.
3) Tools for firmware update, changing boot programs, debugging, etc.
4) Supports proprietary, documented LEGO Communications protocol.

Lego Communication protocol is represented by the structure shown in Figure 6. The following control command format is used according to the Figure:

1) 0-byte: low-order byte of the command;
2) 1st byte: high-order byte of the command;
3) 2nd byte: control command type;
4) 3rd byte: control command;
5) 4th byte: control command argument;
6) 5, 6, 7th bytes: end of frame, command sending.


```
byte [] leftTurnSignal = new byte[]{0x0C, 0x00, 0x00, 0x04, (byte)0xFF, 100, 0x01, 0x00, 0x00, 0x20, (byte)0x80, 0x00, 0x00, 0x00};
byte [] straightSignal = new byte[]{0x0C, 0x00, 0x00, 0x03, (byte)0xFF, 100, 0x01, 0x00, 0x00, 0x20, (byte)0x80, 0x00, 0x00, 0x00};
byte [] rightTurnSignal = new byte[]{0x0C, 0x00, 0x00, 0x02, (byte)0xFF, 100, 0x01, 0x00, 0x00, 0x20, (byte)0x80, 0x00, 0x00, 0x00};
byte [] backwardsSignal = new byte[]{0x0C, 0x00, 0x00, 0x01, (byte)0xFF, 100, 0x01, 0x00, 0x00, 0x20, (byte)0x80, 0x00, 0x00, 0x00};
```

Figure 6: Frame byte sequence

In addition to the above, Figure 6 shows the byte order for "left", "right", "forward", "backward" control commands.

To create an MD-based smart machine vision system, a preliminary review of open source computer vision libraries has been conducted against such criteria as security, quality, reconfigurability, flexibility, controllability, support.

Table 2: Pros and cons of computer vision open source libraries

Name	Pros	Cons
OpenCV	Actively developed Multitude of implemented algorithms Detailed documentation	Android OS deployment complexity Program code in C/C++
CCV	Automatic classification of detected objects	Program code in C/C++
JavaCV	Easy to use on Android OS OpenCV add-on	Poor performance Lack of detailed documentation
simpleCV	OpenCV add-on Detailed documentation	Python implementation Android OS deployment complexity
VISP	Modularity Detailed documentation	ROS OS oriented

Based on the pros and cons of the computer vision libraries and frameworks presented in Table 2, OpenCV and JavaCV proved to be the most suitable tools for recognizing obstacles on the MR's path. The main advantage of OpenCV is the detailed documentation related to developing applications for fixed platforms, as well as for MU. OpenCV is an open source library designed to solve computer vision problems, the development of which started in 1999. The implementation language is C/C++. The library supports such OS as Linux, Windows, Mac OS X. The main idea of such a product is to provide developers with a set of tools and methods for solving computer vision problems at a higher level, while it is worth noting that due to low-level languages used for the library's implementation, such systems will have high performance and speed. The library consists of the following modules, listed in order of importance:

1) Kernel — contains basic data structures, such as multidimensional arrays for storing image data, mathematical solutions and basic functions used by other modules.
2) Image processing module — includes mathematical formulas for image filters, geometric representations of converted images, histograms, etc.
3) Video stream processing module — analyses the video stream, contains algorithms for object tracking.
4) Graphical user interface module — provides means for user interaction with the application for solving computer vision problems.

Connecting OpenCV library to OS allows using the full potential of the library in developing applications for Android mobile OS (Yevstigneev, 2003). Mobile OS resource manager can be used to interact with OpenCV manager, which provides the following advantages:

1) Efficient RAM utilization.
2) Hardware optimization for supported devices.
3) The library can be installed via Google Play service.
4) Automatic and regular library updates with bug fixes.

5. DEVELOPMENT OF MR MOTION PLANNING ALGORITHM

Let us consider three alternatives for an MR motion control system:

1) Distributed decision-making system. Such system implies availability of a server for comprehensive processing of big data, making more accurate decisions on the choice of MR movement, speed and acceleration. The action sequence flow chart is presented in Figure 7, a.
2) Reactive decision-making system. Such system implies quick interaction between the executive and the controlling part of the robot. The advantage is speed, while the disadvantages are lower MR positioning accuracy and high probability of collision with obstacles. The action sequence flow chart of such system is shown in Figure 7, b.
3) Hybrid decision-making system. Such system implies a symbiosis of two previous system types, i.e. having the speed of the reactive decision-making system and the intelligence of a distributed decision-making system. Thus, this system can be divided into three levels as shown in Figure 7, c. The balancing level is responsible for the safe and consistent execution of instructions at each level, as well as their coordination.

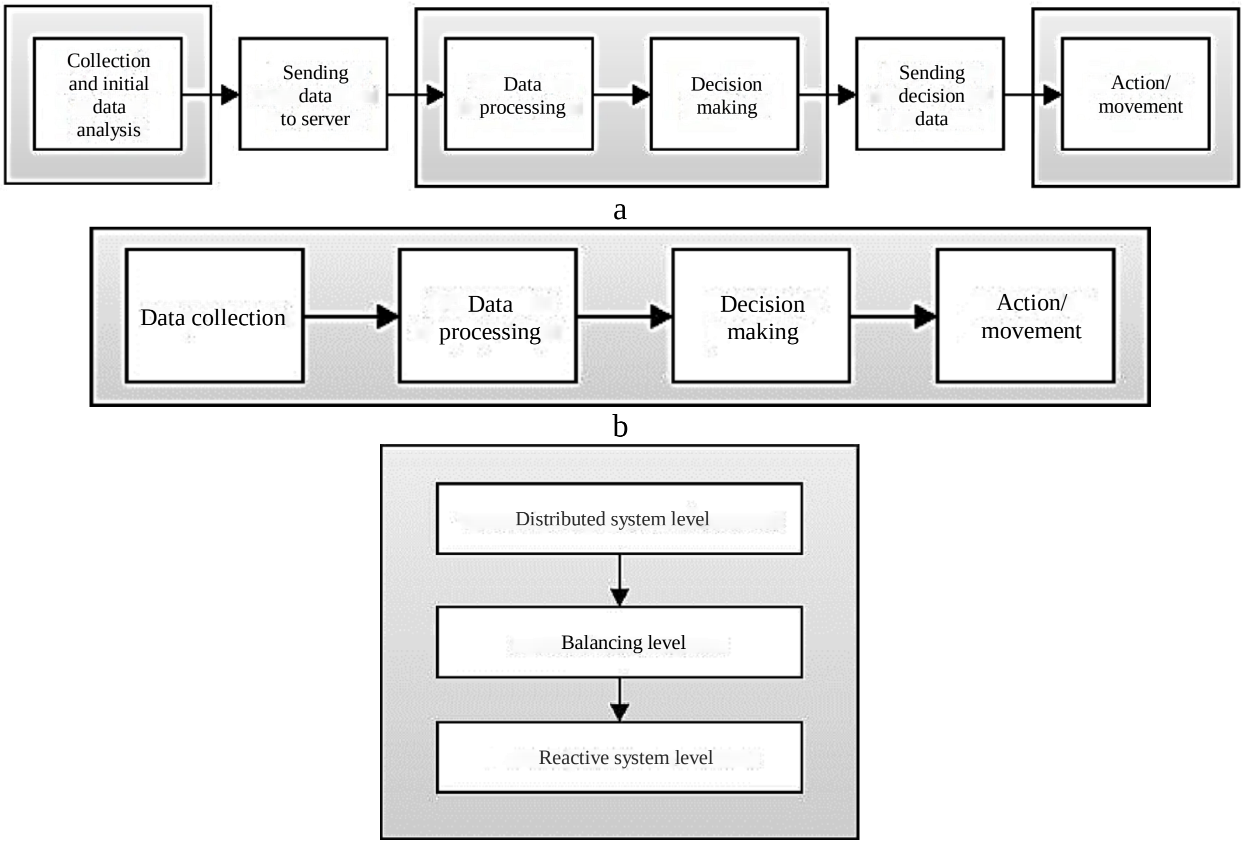

Figure 7: Three alternatives of a motion control system. Action sequence flow chart for a distributed decision-making system (a); Flow chart for a reactive motion planning system; Representation of hybrid system levels (c)

In case MR design has only one camera as a machine vision system to determine the robot's movement direction, the following three sequential tasks shall be completed: identifying an environmental object as an obstacle, short-term tracking of an obstacle to eliminate external short-term disturbances (bright flashes, for example, sun reflections; driving through rough terrain, which makes robot's field of view unstable), sending a recorded frame with an obstacle to the algorithmic image processing unit, and further MR motion planning (Karmanov, 2004).
For image recognition and identification, it is also necessary to implement digital image processing algorithms. The methods used in image processing to solve this problem are:

1) photogrammetry;
2) image quality enhancement;
3) image filtering.

In case the photogrammetry is used when the distance between the camera and the observed object is reasonably bigger than the optical system's focal length, then the image is built in its focal plane (Göbel et al., 2011). The projection of M point can be represented as

$$Zv = AM \quad , \tag{10}$$

Where

$$A = \begin{pmatrix} f/w & 0 & u_0 \\ 0 & f/h & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$ — internal camera parameters matrix,

M — point M three-dimensional vector, v — camera homogeneous coordinates vector.
Linear contrast enhancement method is used to solve the image quality enhancement problem (Göbel et al., 2011). If image pixel brightness values differ drastically from the brightness range thresholds, then the visualized image looks not sharp enough — blurry contours are observed that impede image detection. Element-wise transformation can be represented as

$$y = ax + b, \tag{11}$$

where x, y are the image brightness values before and after processing, respectively, a and b are the transformation parameters.
Combination equations for the minimum and maximum brightness values can be represented as follows:

$$\begin{cases} y_{min} = ax_{min} + b \\ y_{max} = ax_{max} + b \end{cases}, \tag{12}$$

Let us write (12) taking into account the image transformation parameters

$$y = \frac{x - x_{min}}{x_{max} - x_{min}} (y_{max} - y_{min}) + y_{min} \tag{13}$$

Since images formed by various information systems are distorted by noise, automatic computer processing becomes rather complex. Noise reduction can be achieved by filtering, when brightness of each point of the original image distorted by noise is replaced by another brightness value, which is recognized as the least distorted by noise. Therefore, the main idea of filtering is the rational use of data from both the considered point and neighbouring areas (Martyshkin et al., 2018; Martyshkin, 2016). In this lies the significant difference between filtering and element-wise procedures discussed above. Filtering is not an element-wise image processing procedure. This article uses optimal linear filtering based on the Wiener-Hopf equation. The input image is described by the following expression:

$$y_{i,j} = f(x_{i,j}, n_{i,j}), \ \ i = \overline{0, \ I-1}, j = \overline{0, \ J-1}. \tag{14}$$

where: n is the amount of noise at the point, f (x,n) is the function of signal-noise interaction, I and J are the number of rows and columns in the image frame, respectively. In case of linear filtering, the output effect is determined by a linear combination of input data:

$$x^*(i,j) = \sum_{(i_1,j_1)\in S} \sum a(i_1,j_1) y(i - i_1, j - j_1) \tag{15}$$

where x is the desired signal filtering result at a neighbouring point with coordinates (i, j); a(i, j) — weighting factors.
MR motion planning objective is to find the optimal, free path from the starting point to the destination point in an unknown environment containing all kinds of obstacles. Before starting

with algorithm development, we have classified motion planning methods. Since motion planning is an important aspect of MR navigation research and depends on various situations, we have derived the classification presented in Figure 8 during the study.

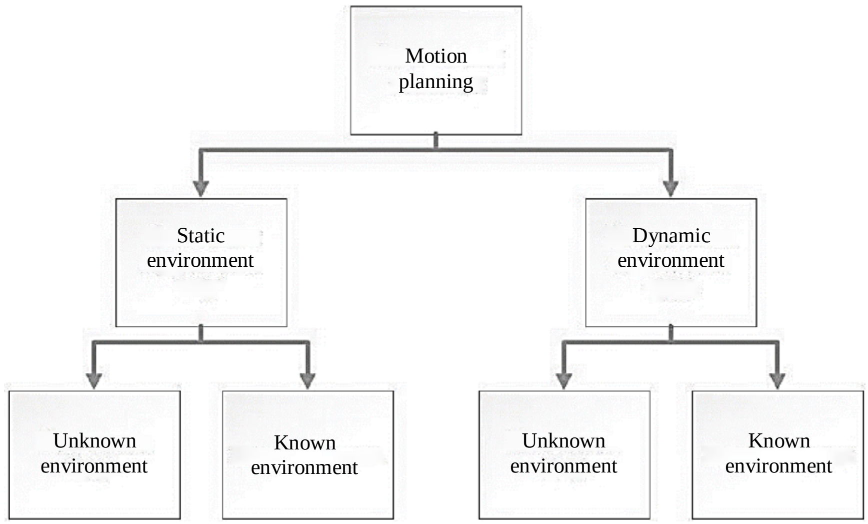

Figure 8: Motion planning process classification

Motion planning can be divided into 2 groups: planning in static and dynamic environments. Static environment can be seen as an environment containing only static obstacles, and dynamic environment — dynamic obstacles. Each of these groups is divided into subgroups: controlling MR in an environment with known map and objects, and unknown map and objects, which shall be identified in real time.

For each case, the most suitable methods for path-finding and motion planning are given below.

1) Motion planning in a known static environment with known obstacles

In this case, the MR receives information about the map of the environment and obstacles before starting to move. Therefore, the optimal path can be calculated before the movement start in the "offline" mode. Planning methods for such environments are sufficiently advanced and are successfully applied in industrial sector. A widely used method is motion planning based on visibility graph analysis, which implies building a segmented zigzag line, which vertices coincide with vertices of the obstacles on the map, if there is a geometric representation of the obstacle map, starting and ending points of the path.

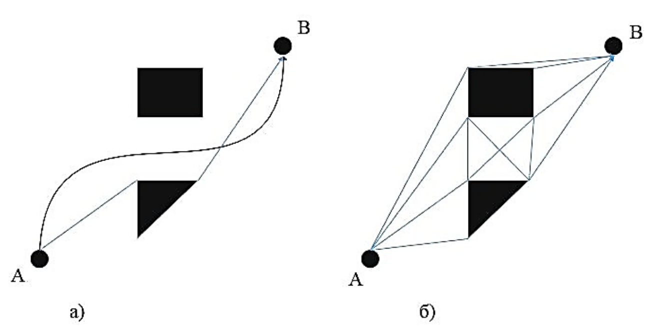

Figure 9: Visibility graph

Figure 9 shows a geometric map of the area with two obstacles, path starting point A and ending point B. Figure 9a shows the optimal path and the segmented zigzag line of the path. Figure 9, b shows all possible paths, the optimal choice can be set according to custom criteria for solving specific problems.

2) Path planning in a static unknown environment

In this case, the task is complicated by the impossibility of preliminary analysis and solving the optimal path planning problem. To solve this problem, utilization of local navigation means in real time shall be maximized, given the heterogeneity of the obstacles located in the MR's operation area. In the course of the study, possible situations arising from MR movement have been considered (Bradski & Kaehler, 2008). Most of these situations and their solutions are presented in Figure 10.

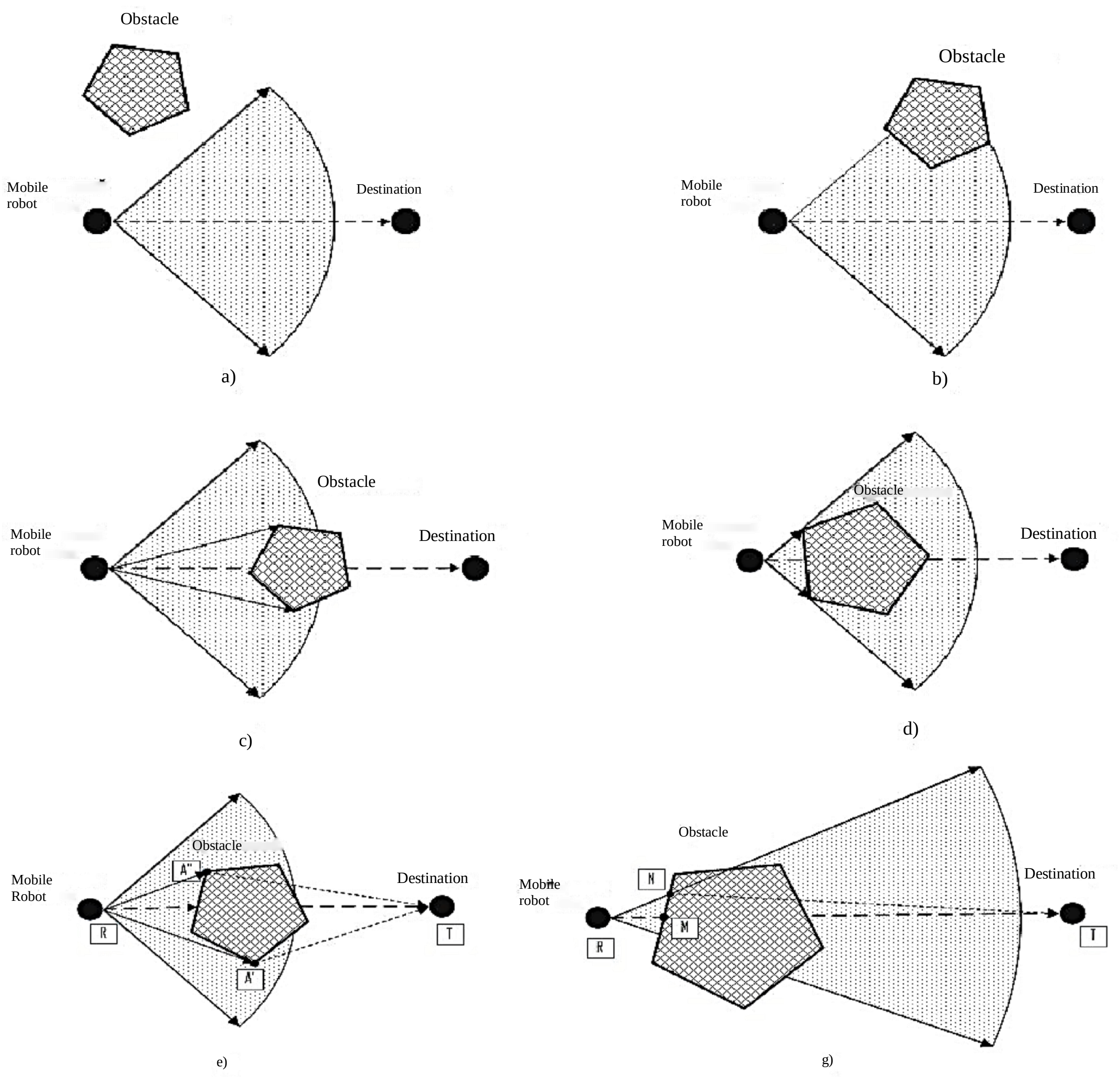

Figure 10: Geometric positions of a mobile robot relative to an obstacle

The initial data in this case are the static terrain with unknown obstacles, MR, path start and end coordinates. A detailed description of the relationship between MR and an obstacle is provided below:

a) No obstacles are detected by MR in the scanned environment on a straight path to the destination. MR reaches the destination moving in a straight line.

b) An obstacle of an arbitrary geometric shape is detected by MR in the scanned environment on a straight path to the destination, which does not intersect with the planned trajectory of movement. MR reaches the destination moving in a straight line.

c) An obstacle of an arbitrary geometric shape is detected by MR in the scanned environment, which intersects with a straight path to the destination. MR cannot move straight towards the destination. Further movement towards the destination is only possible after obstacle avoidance algorithm execution.

d) An obstacle of an arbitrary geometric shape is detected by MR in the scanned environment, which intersects with a straight path to the destination and completely obstructs the MR's field of view. MR cannot move straight towards the destination. Further movement is possible after executing the data collection algorithm and then — the obstacle avoidance algorithm.

Considering that cases c) and d) do not allow MR to reach the destination, possible options for reaching the destination were considered. For cases c) and d) the solution is shown below:

e). The planned path is calculated from the current MR position R in the form of straight lines RA'T + x and RA''T + x to the extreme points A' and A'' of the obstacle with the corresponding offset x. The minimum offset shall be greater than the half of the MR width. Choice of the maximum offset is based on the preliminary knowledge about the MR's operating environment. Having reached intermediate destinations at points A 'or A'', MR executes the algorithm recursively until it reaches the final destination.

f) The planned path is calculated only after the MR makes a turn in-place, thereby expanding its field of view, moving point M to N. After that, the algorithm described in case e) is executed.

3) Path planning in a dynamic known environment

Based on the practical experience of many applications, MRs often encounter dynamic obstacles. The coordinates, speed and acceleration of such obstacles are usually constantly changing. Keeping these statements in mind, it can be concluded that motion planning in a known environment with dynamic obstacles is a complex task, which requires ultimate tracking of dynamic obstacles, as well as forecasting the obstacle's coordinates, speed and acceleration, to be effectively solved. It is worth noting that the complexity of this problem increases in a proportion to the number of dynamic obstacles. The path planning algorithm calculates a sequence of steps between positions located on a map of the environment, divided into a grid, for navigation to the destination, taking into account known static and moving obstacles with predictable trajectories (Montemerlo & FastSLAM, 2007). Three cost functions are used:

a) The first function calculates the step direction relative to the current position in the direction of cells, which will be free from obstacles in the upcoming period of time.

b) The second function calculates MR movement cost into the next map cell.

c) The third function calculates the residual cost of movement to the destination, which is calculated in the opposite direction from the destination. Introducing this function in the motion planning system helps to avoid local lows.

4) Path planning in a dynamic unknown environment

MR motion planning in an unknown environment with static and dynamic obstacles is the most difficult case, but at the same time the most common one encountered by MRs. So, in a real-world example, unmanned underwater vehicles need to avoid collisions with other vessels, but also with large marine animals. In a difficult unknown environment, MR cannot plan a global path to the destination from the beginning and at once. MR continuously uses sensors along its path to obtain information about the environment, continuously analyses the obtained data in real time, plans and adjusts the path in accordance with the algorithms embedded in the software. The time of each operation should be minimized as much as possible to ensure acceptable quality indicators, such as positioning accuracy, average movement speed along the entire path and time to reach the destination. The most effective method in this case is the potential field method proposed by A.K. Platonov in 1970 (Bradski, 1998). The necessary conditions are high accuracy of the MR navigation system, known initial and final MR coordinates, direction of the field of view in the selected coordinate system. In this coordinate system, MR is represented as a point with an orientation vector. The essence of the method is described below. The destination is positively charged, and the MR and obstacles are negatively charged. Influenced by forces, MR having a negative charge is attracted to the positively charged destination, and repelled from negatively charged obstacles. This method is essential because it allows setting your own laws of motion in many different ways. Consequently, with the introduction of restrictions on the unknown environment and the laws of MR motion, MR will reach the goal.

The developed obstacle avoidance algorithm is presented in Figure 11. In a known static environment with unknown coordinates of obstacles, in the simplest case with no obstacles on the MR path, MR reaches the destination without collisions with minimal time overheads. In other cases, additional conditions are met, and a virtual probability collision sensor is introduced into the system. The acceptable noise level as well as the number of seconds N required to capture and track an obstacle is configured during MR setup and calibration. When a obstacle is detected, i.e. when an object is identified as an obstacle, the value of the MR sensor is incremented, then the MR speed is reduced by 50%, and the obstacle is tracked for N seconds. If tracking has been successful for N seconds, the collision probability sensor value is reset to zero, and MR is turned by a calculated angle. In the opposite case, the collision probability sensor value is incremented, if it is not exceeding the threshold K, which is also set during MR setup and calibration stage. If the collision probability sensor value exceeds the threshold K, MR immediately stops, sending a signal to the server, specified before the movement is commenced, with a message about the complex situation on the path. The received response may imply switching MR to the manual control mode, or contain processed conflicting image frames.

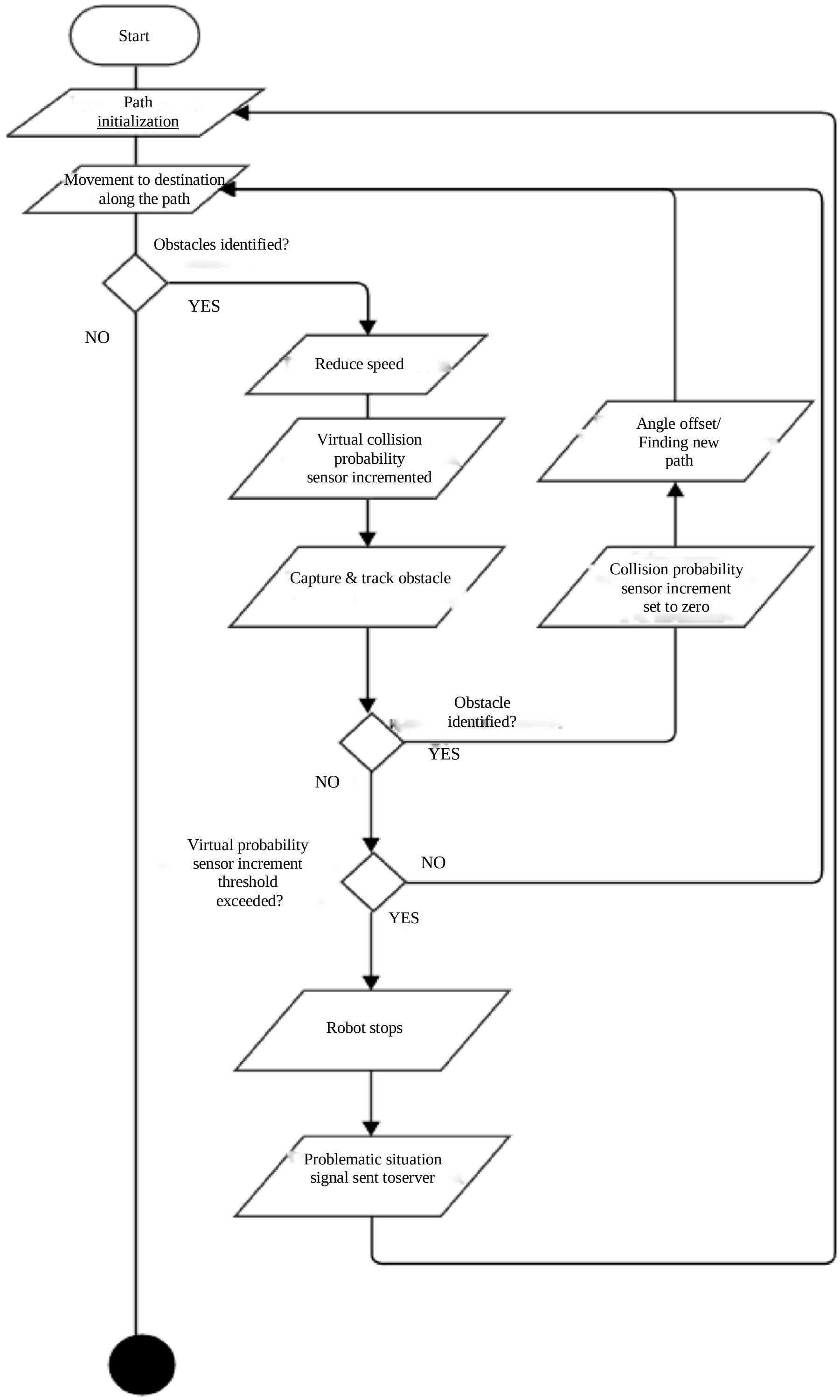

Figure 11: Obstacle avoidance algorithm for a mobile robot

6. AN EXPERIMENTAL STUDY OF THE DEVELOPED MOTION PLANNING SYSTEM

To conduct a practical experiment using virtual means for MR motion algorithms testing, a review of the existing software solutions for simulation of such systems has been conducted. The benefits of preliminary testing of MR algorithms include:

1) Solving the problem of limited choice of hardware and software. Virtual MR simulation environments allow assembling robots from the most relevant and commercially available units.
2) Inability to monitor robot behaviour in certain environments, most often aggressive ones.

The main features of robot simulation software are:

1) parallel and asynchronous processing of data coming from sensors;
2) synchronous and asynchronous interaction between control processes and various robot subsystems;
3) robot model creation taking into account physical properties, such as mass, elasticity, material properties, torques;
4) testing of control algorithms;
5) creation of customizable (including physical and geometric parameters) 3D scenes for simulation;
6) ability to simulate individual MR assemblies, for example, modern sensors: laser rangefinder, ultrasonic and infrared sensors, compass, drives and servo motors, GPS, etc.

An easy-to-use Webots software package has been chosen to simulate the algorithm created during the study. Despite the fact that Webots is a paid software, it has a trial period of up to three months. Webots is great for robot simulation tasks, being quite easy to learn due to detailed documentation in English on the product's official website. It has built-in support for several functional and object-oriented programming languages. Figure 12 shows the simulation scene in the form of an oval arena, a box-shaped obstacle, and MR, which shape is simplified due to limitation of standard components of the software package. By opening Sensors tab, the user can get immediate data continuously coming from the sensors in real time. At the end of the experiment, statistics is available. The objective of the experiment is MR reaching the destination point, which is located on the opposite side relative to the starting point of movement.

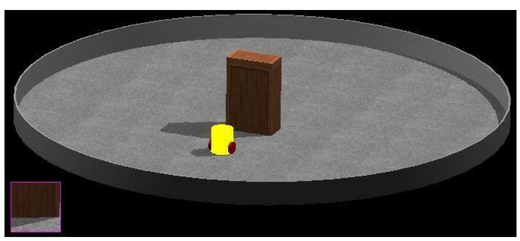

Figure 12: Mobile robot and environment simulation scene

During MR prototype testing in the natural environment, the same algorithms for obstacle detection and avoidance have been used. MATLAB math toolbox has been used to create graphs based on MR coordinates that show the MR path. Figure 14 (a) shows the obstacle avoidance path for one obstacle in the environment. Figure 14 (b) shows the obstacle avoidance path for two obstacles. Figure 14 shows the obstacle avoidance path for multiple obstacles in the environment.

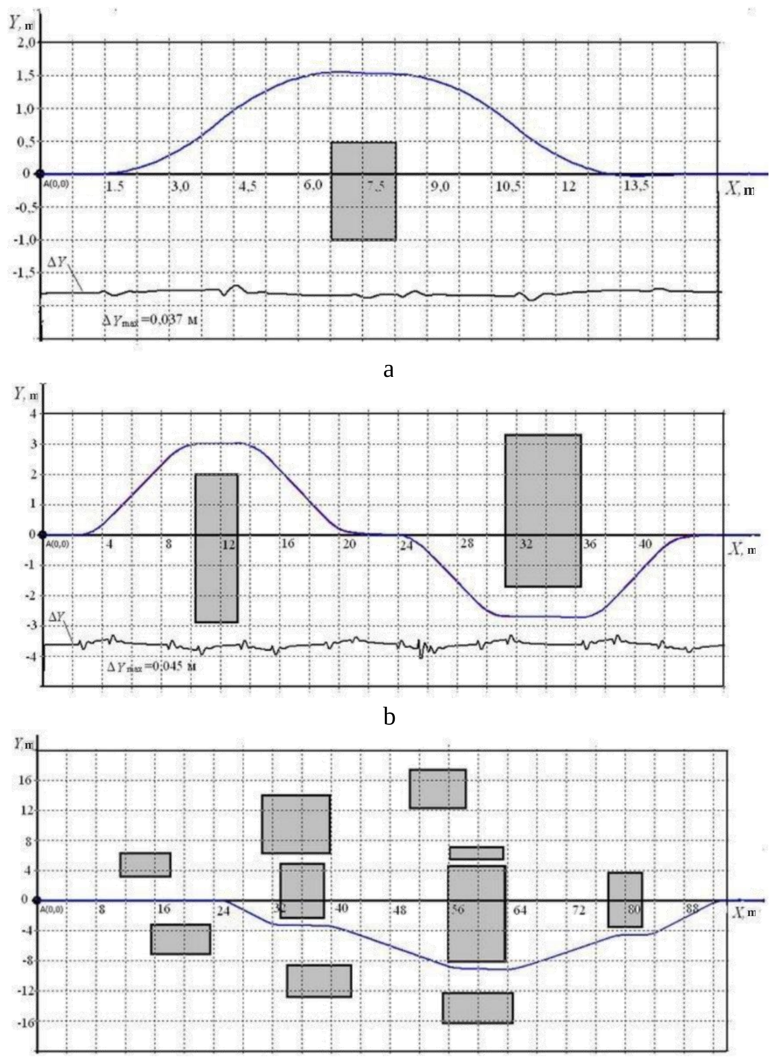

Figure 14: Movement trajectory on a obstacle avoidance path around one obstacle (a), two obstacles (b), multiple obstacles (c)

Also, Figure 15 shows an image received while tracking an obstacle. Such tracking is ranges from several milliseconds to several seconds before MR decides on creating a new movement path to reach the destination and turning by an angle ψ. Many points that are not centres of the highlighted contours are temporary noise. It can be caused by both sun reflections and the lack of a camera stabilization system. To solve this problem, the video camera should be outfitted with a stabilization system. Stabilization will allow to achieve higher performance in terms of obstacle recognition speed, increased noise immunity, and MR positioning accuracy in the environment.

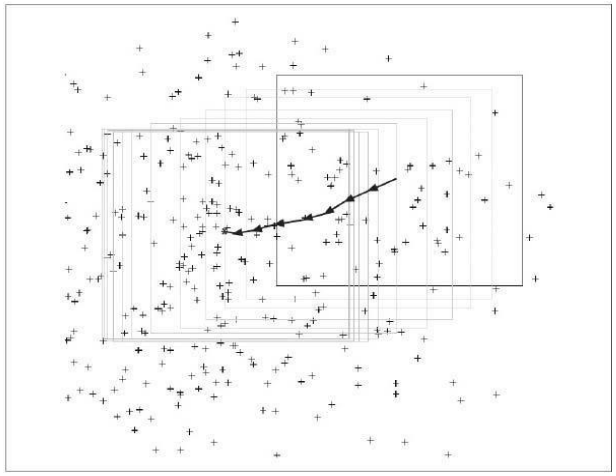

Figure 15: Tracking the obstacle contour

The result of the experiment, which is made possible by joint use of sensors of the designed MR and obstacle avoidance algorithm execution, is MR reaching the specified destination without touching the obstacle.

7. CONCLUSIONS

This study provides a review of domestic and foreign MR that solve the motion planning problem in a known environment with unknown obstacles. Such MR navigation methods as local, global and individual are considered. The hardware components have been selected including MD running Android 5.0 OS and the actual executive part of the device based on a three-wheeled platform Lego Mindstorms NXT with two driving servo motors and a castor wheel. Based on the research concluded, a prototype MR has been built, capable of recognizing obstacles of regular geometric shapes, as well as performing motion path planning and correction, if required. Environment objects are identified and classified as obstacles by means of digital image processing methods and algorithms. Distance to the obstacle and relative angle are calculated by photogrammetry methods, image quality is improved by linear contrast enhancement and optimal linear filtering using the Wiener-Hopf equation.
Virtual tools, related to MR motion algorithm testing, have been reviewed, resulting in selection of Webots software package for prototype testing. Testing results allowed us to make the following conclusions. The mobile robot has successfully identified the obstacle, planned a path in accordance with the obstacle avoidance algorithm, and continued moving to the destination.

ACKNOWLEDGMENT

The article is published with the support of the scholarship from the President of Russian Federation to young scientists and graduate students in 2018-2020 (SP-68.2018.5).

REFERENCES

Zavlangas, P. G., Tzafestas, S. G., & Althoefer, K. (2000). Fuzzy obstacle avoidance and navigation for omnidirectional mobile robots. In Proceedings of the third European Symposium on Intelligent Techniques.

Chi K.-H., & Lee M.R. (2011). Obstacle avoidance in mobile robot using Neural Network. International Conference on Consumer Electronics, Communications and Networks (CECNet), 5082–5085.

Kristensen, S., Horstmann, S., Klandt, J., Lohner, F., & Stopp, A. (2001). Human-friendly interaction for learning and cooperation. Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea, 2590-2595.

Ulas, C., & Temeltas, H. (2013). Multi-Layered Normal Distribution Transform for Fast and Long Range Matching. Journal of Intelligrnt & Robotic Systems, *71*(1), 85-108.

Voitovich, I.D. (2009). Smart sensors. I.D. Voitovich, V.M. Korsunsky // Publishing house: "Binom. Knowledge laboratory", 51–53.

Dergachev, V.V. (2017). Methods for assessing and reducing the computational complexity of decision-making algorithms in problems of simultaneous positioning and mapping / V. V. Dergachev, O.O. Kartashov. Engineering Journal of Don, *4*.

Kuchersky, R.V. (2012). Algorithms of local navigation and cartography for the on-board control system of an autonomous mobile robot / R.V. Kuchersky, S.V. Manko // SFEDU News. Engineering sciences.

Montemerlo, M., & FastSLAM, S. T. (2007). A Scalabale Method for the simultaneous localizaton and mapping problem in robotics. Springer Tracts in Advanced Robotics, *27*.

Visilter, Yu.V. (2007). Processing and analysing digital images using LabVIEW IMAQ Vision with examples. DMK Press.

Lokhin, V.M., Manko, S.V., Romanov, M.P., Alexandrova, R.I., & others. (2008). Multipurpose on-board control system for autonomous mobile weapons and military equipment. Proceedings of the III scientific & practical conference "Perspective systems and control challenges". V. 1. - Taganrog: Publishing house of TTI SFEDU.

Romanov, M.P., & Gartseev, I.B. (2008). Smart navigation system for small-sized mobile weapons and military equipment. Proceedings of the III scientific & practical conference "Perspective systems and control challenges". V. 2. - Taganrog: Publishing house of TTI SFEDU.

Castellanos, J.A., Neira, J., & Tardos, J.D. (2004). Limits to the consistency of the EKF-based SLAM. In Intelligent Autonomous Vehicles (IAV) (Lisboa, PT).

Durrant-Whyte, H. (2002). Localization, Mapping, and the Simultaneous Localization and Mapping Problem. Australian Center for Field Robotics. Sydney.

Montemerlo, M. (2003). A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association. Ph. D. thesis, Carnegie Mellon University.

Nist´er, D., Naroditsky, O., & Bergen, J. (2006). Visual odometry for ground vehicle applications. Journal of Field Robotics, *23*(1), 3-20.

Devyaterikov, Ye.A., & Mikhailov, B.B. (2015). Use of visual odometer data for autonomous return of a mobile robot in an environment without fixed reference points. Extreme Robotics: Collection of reports of the All-Russian Scientific and Technical Conference. - St. Petersburg: Publishing House "Polytechnika-Servis", 356-361.

Gerasimov, V.N., & Mikhailov, B. B. (2012). Solving the problem of mobile robot motion control in the presence of dynamic obstacles. Bulletin of MSTU named by N.E. Bauman. Instrument engineering. Special issue "Robotic systems", *6*, 83-92.

Gerasimov, V.N. (2015). SLAM algorithm based on the correlation function // Extreme Robotics: Collection of reports of the All-Russian Scientific and Technical Conference. - St. Petersburg: Publishing House "Polytechnika-Servis",126-133.

Gerasimov, V.N. (2014). Regarding the issue of controlling mobile robot movement in a dynamic environment. Robotics and engineering cybernetics, *1*(2), 44-51.

Thrun, S. (2004). Fast SLAM: An Efficient Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association / S. Thrun [et al.] // Journal of Machine Learning Research.

Calonder, M. (2006). EKF SLAM vs. FastSLAM-A Comparison. Article Computer Vision Lab. Swiss Federal Institute of Technology, Lausanne (EPFL).

Bradski, G.R. (1998). Computer vision face tracking for use in a perceptual user interface.
Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc."

Fukunaga, K., & Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on information theory, *21*(1), 32-40.

Göbel, S., Jubeh, R., Raesch, S. L., & Zündorf, A. (2011). Using the Android platform to control robots. Kassel University Germany.[Online]. Available www. innoc. at/fileadmin/user_upload/_temp_/RiE/Proceedi ngs/65. pdf.

Plotnikov, V.A. (2010). Analysis of the effectiveness of existing collision avoidance methods for mobile robots. Donetsk: Artificial intelligence.

Strom, J., Slavov, G., & Chown, E. (2009). Omnidirectional walking using zmp and preview control for the nao humanoid robot. RoboCup*:* robot soccer world cup XIII.: Springer*,* 378–389

Yevstigneev, D.V. (2003). Software and algorithmic support for intelligent control systems of autonomous mobile robots: thesis abstract ... for PhD in Engineering: 05.13.01 / Moscow State Institute of Radio Engineering, Electronics and Automation. — Moscow,. — 26 p.

Karmanov, V.G. (2004). Mathematical programming.- 5th ed., stereotypical. - Moscow: Fizmatlit, - 263 p.

Martyshkin, A.I., Salnikov, I.I., Pashchenko, D.V., & Trokoz, D.A. (2018). Associative Co-processor on the Basis of Programmable Logical Integrated Circuits for Special Purpose Computer Systems. Proceedings - Global Smart Industry Conference, GloSIC 2018, art. no. 8570067. doi: 10.1109/GloSIC.2018.8570067

Martyshkin, A.I. (2016). Mathematical modeling of Tasks Managers with the strategy of separation in space with a homogeneous and heterogeneous input flow and finite queue. ARPN Journal of Engineering and Applied Sciences, *11*(19), 11325-11332.