

# Autonomous navigation system with small four-wheel drive platform

*Martin Bogdanovskiy\**, *Andrii Tkachuk*, *Oleksandr Dobrzhanskyi*, and *Anna Humeniuk*

Zhytomyr Polytechnic State University, Department of Automation and Computer-Integrated Technologies, Zhytomyr, Ukraine

**Abstract.** The task of achieving greater flexibility and maneuverability of small transport and service units' motion in modern factories by developing small autonomous navigation systems plays crucial role in complex automation of transport logistics nowadays. To solve navigation task, it was proposed the following approach, where as a means of assessing the environment was used computer vision system based on 5-megapixel CMOS image sensor and for the front obstacle detection was used auxiliary ultrasonic sensor as a limit switch. Authors solved the problem of yawing using artificial marking approach as along two-colored leading lines. For maneuverability increase during the turn was used speed movement control based on lines perspective. The basic design and technical characteristics of the four-wheel drive platform and the algorithm of the Raspberry PI 3/Arduino Nano hybrid control system are presented. Experimental results proved the viability of the proposed approach.

## 1 Introduction

According to last report of Grand View Research company [1], the global Automated Guided Vehicle (AGV) market is dynamically growing. This is thanks to the rising demand of automation systems based on computer vision technology. This type of systems has found wide application in transport logistics, assembly, packaging and storage facilities. The automotive, food, aerospace, pharmaceuticals and retail sectors most of all need this technology to make the service operation process safer for personal and other co-operating technological equipment. AGV gives ability to increase productivity and minimize operational costs by using Industrial 4.0 technologies workspace organization [2]. The automation equipment offers a number of classic and innovative solutions to reach high flexibility with approved cost-effective techniques in implementation. In addition to large systems such as robot platforms, small transport and service units are capable to perform path observation, complete simple operations as monitoring and, in some cases, can be combined into a cluster to increase the availability by spatial configuration with one or several control units.

Rapid development of computer systems and workspace recognition techniques, together with the increase of units' number working together and the decrease of their cost, stimulate research of new and combined solutions based on the available hardware and software systems. Smart sensors give opportunity to analyze working space by several spots with common dataset. The use of modern automation systems and

technologies in navigation allows us to introduce new approaches in solving these tasks.

## 2 Formulation of problem

Improving the efficiency and functionality of small transport units for production and service includes a wide range of tasks such as transport logistics optimization, autonomous navigation and path planning, stability of the motion characteristics in a loaded / unloaded state, safety of transport and service operations, distribution of computational power among control units etc. The workspace is generally characterized by growing the intensity of traffic flows with a large number of situations which require rapid response. Finding a solution is usually a compromise between the complexity and reliability of the system and the ability to real-time reconfigure the base route in the workspace.

## 3 Literature review

Under the conditions of small-volume flexible production, small autonomous transport systems are in the wide demand. They give the possibility of monitoring or transporting small parts in trays or carriages in limited workspace. The popular techniques for navigation use magnetic and colored tapes, light reflective tapes with laser scanning, radio spots, computer vision system and systems with inertial adjustment of motion. Some important works in this field devoted to use special techniques of autonomous navigation.

The paper [3] describes the use of magnetic and color bands, light reflecting bands with laser scanning, radio

\* Corresponding author: [martin.bogdanovskiy@gmail.com](mailto:martin.bogdanovskiy@gmail.com)

beams, systems of vision and inertial motion control for the construction of navigation systems. However, the problem of recognizing objects that may intersect a route (according to the trajectory of the object) is not resolved.

In the paper [4], the authors propose a combination of an ultrasonic sensor with a computer vision system to accurately determine distances to the frontal obstacle. However, the proposed system is used for free route planning and is very complex in data preparation for recognition in advance.

In papers [5, 6], the authors propose free-motion systems based on lidar, capable of recognizing the surrounding environment for maneuvering. However, the sensors proposed in the papers provide limited detail for interacting with objects using color techniques.

The works [7, 8] are devoted to autonomous navigation along a fixed route, constructed using proportional-integral-differential (PID) and neuro-fuzzy control laws, mainly based on deterministic and / or simulated data sensors. However, the works do not pay attention to the fact that in the case of computer vision, a sufficiently complex filtering and algorithmic adaptation to the workspace is required.

Articles [9, 10, 11] describe modern sensors for measuring acceleration and distance to objects. However, they do not indicate the feasibility of using them in the design of autonomous mobile platforms. The possibility of replacing them with artificial marking systems has not been indicated.

Using the computer vision system gives good opportunity to estimate a lot of workspace parameters for navigation. However, the sensor work is affected by a variety of environmental conditions, on which the result of recognition depends. One of the main recognition problems is the uniformity of illumination and the presence of glare surfaces. Even in the conditions of artificial lighting, brightness level changes as a result of the appearance and disappearance of objects with different light reflecting and light absorbing properties. Additional artificial marking of routes like colored leading lines and markers can also simplify or improve the navigation task, but the problem of objects recognition which cross or can cross the route (according to trajectory of object movement) is still present. So, the ability to recognize objects and obstacles in workspace variably to each situation is an intellectual task.

The use of powerful embedded computers and flexible software tools gives an opportunity to solve complex data processing tasks on small autonomous mobile platforms.

## 4 Proposed work

The aim of this study is to implement an autonomous navigation system based on computer vision and ultrasonic sensor that are placed on a small four-wheel drive (4WD) platform as a prototype of AGV. The tasks considered in this paper include electric circuit design and control implementation taking into account raw data filtering along the path. Solving them will let proposed small 4WD platform autonomously navigate along a defined route with the future ability to detect the

obstacles. In our study we define the following assumptions: 1. the workspace is a closed facility area; 2. navigation task includes preliminary determined location, geometric parameters of technological equipment and the artificial lighting scheme of the workspace.

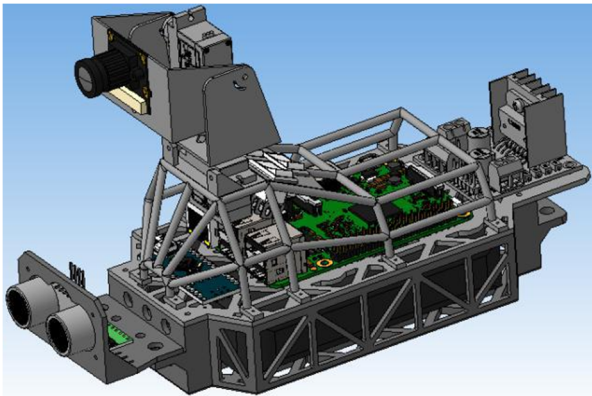
### 4.1 Hardware implementation

All sensors are installed directly on the 4WD platform to ensure high mobility and reliability of autonomous navigation within the closed workspace. The main navigation task is to provide unhindered movement in the forward direction. For leading lines and objects detection, the computer vision system includes 1080p high resolution monocular CMOS camera with a 160° viewing angle and a variable focal length based on the chip OV5647. The data from 3-axis accelerometer and gyroscope of the MPU-6050 module are used to get a stabilized video stream and to hold the camera in a horizontal position. To avoid collisions in unpredictable situations such as incorrect objects recognition of the workspace, the wheeled platform has an additional ultrasonic distance sensor HC-SR04, which has no “blind zones” in the front plane with a measuring range from 0 to 150 cm and with an accuracy of 0.3 cm.

Estimation of the workspace, due to the dynamic characteristics of the wheeled platform and surrounding objects, requires the rapid processing of video data received from the camera. A single-board computer Raspberry Pi 3 provides enough computational power for processing video stream and to control engines and servos. Raspberry Pi 3 implements standard interfaces for wire and wireless communication for remote control, and also supports high-level programming languages. To provide maneuverability, the mobile platform uses a classic steering system. Taking into account small dimensions and model weight, a small SG90 servo actuators are used for steering the platform, providing a sufficient starting torque of 2 g/cm. To achieve required dynamical characteristics of the platform authors used DC motor HRK-370SH, which provides a torque of 79.6 g/cm at a starting current up to 28 A. As a driver we selected VNH2SP30 which generates power signals to the drive system. Alignment of sensors and elements of the control system is made with taking into account the minimum length of wires connections and the usability to work with charging and programming interfaces. Creating a control system prototype for further scaling to a physical AGV needs to take into account the type of wheelbase of the real system. Therefore, wheelbase was chosen from ready-made solution, which provides similar adhesion to the surface and the area for the placement of the control, actuators, sensors and auxiliary elements. An independent cushioned suspension gives ability to reduce the vibrations that affect camera and MPU to the acceptable level. The rigid frame, was designed in KOMPAS-3D and SolidWorks and finally PLA printed, has good weight balance thanks to effective arrangement of the functional elements and battery pack. The communication and structural possibilities to plug in and fast modification of sensors, power and control units in 4WD platform design

was taking into account as well.

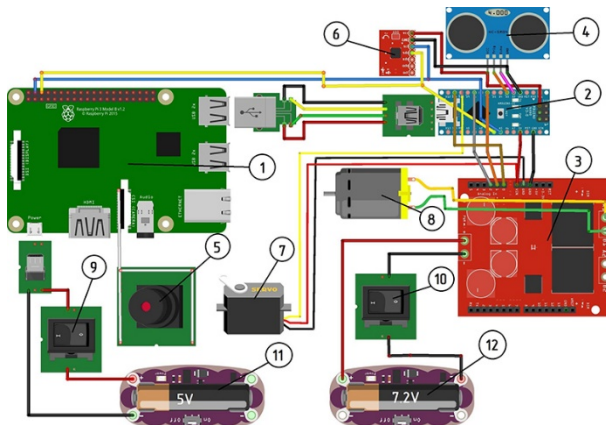
The final design of the assembled control unit is shown on Figure 1.



**Fig. 1.** The general view of control unit with the placed components.

The electrical circuit uses TWI to connect Arduino Nano and MPU-6050 to Raspberry Pi 3. Driver VNH2SP30 connection to Arduino Nano uses standard PWM driver control. Arduino Nano gets power supply through the USB port.

The electrical scheme of connecting the control unit to the drive mechanisms of the wheelbase and sensors is shown on Figure 2.



**Fig. 2.** The electrical scheme of control units and peripheries connections.

The scheme has following items: Raspberry Pi 3 single-board computer (1); Arduino Nano to control driver and ultrasonic sensor (2); VNH2SP30 driver for the main engine (3); HC-SR04 ultrasonic sensor (4); Raspberry Pi OV5647 camera (5); MPU-6050 module with 3-axis accelerometer and gyroscope (6); SG90 servomotor of the steering axle (7); HRK-370SH electric motor of the main movement (8); Raspberry Pi 3 power button (9); HRK-370SH power supply button for the electric motor (10); battery 5 V, 8800 mAh for power supply of electronic circuits (11); Battery 7.2 V, 1500 mAh for main power supply (12).

## 4.2 Control implementation

For safe motion stop of the 4WD platform, the control

system implements the braking mode by calculating of the speed variation which is inversely proportional to the residual distance to the obstacles. The proportional (P) control implements the rigid gear of a controlled value through feedback over distance, determined by the following equation:

$$\mu = K_p \cdot \Delta U_s, \quad (1)$$

where  $\mu$  – is the influence of the regulator on the end-effector (EF) by actuator;  $\Delta U_s$  – is the error signal from the comparison element (CE).

The transfer coefficient  $K_p$  is numerically equal to EF offset when the output value is changed to the unit of measurement. To calculate the P-gain for 8-bit PWM, the control signal range varies from 0 to 255. To overcome the dead zone determined by the forces of dry friction in the wheelbase and transmission, the range of PWM control signals is experimentally defined and narrowed to an interval of 130 to 255.

After the physical experiment, safety zone that guarantees obstacles avoiding by taking into account the massive inertia characteristics of the current platform was determined at the distance of 150 cm. The braking system is calibrated to this level as a reference point to start braking process. Parameter estimation of braking process that is given to the driver is shown below:

1. To set the braking system reference point we subtract the current distance to the obstacle archived by the ultrasonic sensor from safely distance:

$$D_{dist} = 150 - D_{deadzone}, \quad (2)$$

where  $D_{dist}$  – the measured distance to the obstacle in the sensor measurement system;  $D_{deadzone}$  – sensitivity sensor threshold corrected to the safety shift of platform.

2. The driver PWM parameter calculated as:

$$P_{pwm} = 255 - (255 \cdot (D_{dist}/150)), \quad (3)$$

where  $P_{pwm}$  – is the calculated value of the PWM parameter for main actuator power control.

The need to simplify the information contained in the video stream from the high-resolution camera requires filtering. The Open Computer Vision Library (OpenCV) has an application level program interface (API), compatible with high-level programming language and provides interoperability for the implementation of computer vision system [13]. The camera's maximum resolution is  $2592 \times 1944$  pixels. For 2 lines navigation task with stripes wider than 5 cm, the experimentally estimated low threshold of image area is  $800 \times 600$  pixels to operate. Image processing and filtering takes place at several stages [14, 15]. The recognition technique begins from preliminary processing for better markers color detection and vectorization of the input image for mathematical calculations. The camera calibration needs manual tuning at the beginning in order to be able to take into account non-uniform light and color temperature. The color transformation converts blue-green-red color scheme (BGR) to hue-saturation values (HSV) to create the appropriate binary mask for target color of leading lines. Implementation of the Hough transformation [15]

helps to combine the lines to polygons formation.

The next step is to reconstruct the conditions of real workspace to analyze the efficiency of library techniques and navigation parameters. The non-uniform light leads to deviation of contrast of the images, there are false lines for the matching of floor plates. The glare of light prevents interference in leading lines detection, the marking lines have a wavy shape and a heterogeneous structure. After the initial processing and filtering of the source video, the procedure generates an array of coordinate pairs of two points  $(x_1, y_1, x_2, y_2)$  of the recognized polygonal segments of lines. To determine the actual lines, there is a pixel analysis of the fragments of the polygon lines. Each leading line is split to pixels by searching the line neighboring points coordinates. Using Python language [14], the pixel to line normalization is performed using eachPixelOfLine function. The pixel cropping function of the line is implemented with the use of the yield generator. This function provides more efficient use of memory through the processing of one line in one pass and increases the productivity of the analysis. After obtaining the points coordinates, the nearest points to the center of the lines can be found by using the get TheNearestLine function. The points coordinate values in the processing are not constant and Kalman filter [12, 14] helps to average them.

To navigate the 4WD platform, we need to find a horizontal center between the lines, which will be the reference for the adopted steering control. The center estimation is based on the following equation:

$$P_C = (P_L - P_R / 2) + P_L, \quad (4)$$

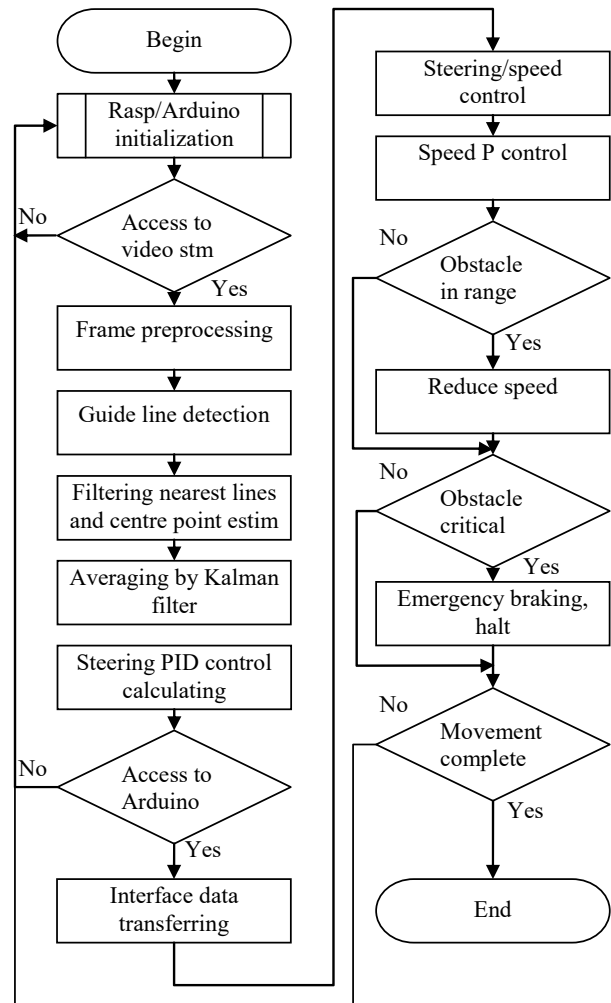
where  $P_C$  – the calculated horizontal center between the recognized lines;  $P_L$  – the nearest left point;  $P_R$  – the nearest right point.

The information about difference between the horizontal center between lines and physical camera orientation relative to the wheels helps to orient the platform on the track, using the PID-control to steer.

The general algorithm of the 4WD platform control system is shown on Figure 3.

After initializing Raspberry PI 3 and Arduino (downloading the created program, importing libraries, defining global variables) the boards verify access to the video stream. Each frame in iteration cycle is undergoing preprocessing. The image is transformed into colorless format for a more efficient procedure of locating leading lines and using openCV library. To achieve the reference point for steering axis the procedure detects center point of tracking path. After implementation of procedures for finding the closest lines the Kalman filter helps to average the central points coordinates. On the next step executes gathering of the actual data for steering, the program checks access Arduino Nano serial interface and transmits the data. During data transition, the terminal activates Arduino Nano that controls driver and ultrasonic sensor connections. To work with peripherals the board initiates the base configuration, variables and libraries. The program calculates the wheel angle and applies it to the driver of steering servomotor. An emergency brake system based on an ultrasonic sensor is triggered if the

distance to the obstacle is less than 30 cm. In this case, the 4WD platform does not move until the obstacle is present. If the distance to the obstacle is in the range of 30 to 150 cm, the speed of the wheeled platform varies in proportion to this distance by P-control.



**Fig. 3.** The general algorithm of the 4WD platform control system.

The 4WD platform navigation software was developed using ST programming language Python3 (for Raspberry PI) and C (for Arduino Nano). Software for Raspberry Pi was developed in JetBrains PyCharm IDE [14]. Searching the artificial leading lines is procedural, with a call to external events. Arduino IDE environment uses procedure submissions with objects' environment model and abstract variables to work with system control like direction, speed, steering angle and brake.

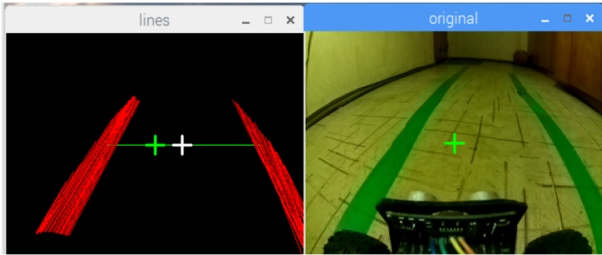
## 5 Experiment

The experimental part of the study is to conduct a natural experiment, which involves the distance movement of 4WD platform along a straight path. Moreover, the nature of surface illumination and the surface itself are complex in terms of reflective properties. The recognition technique involves the use of a pair of green navigation lines, the distance between which is greater than the width

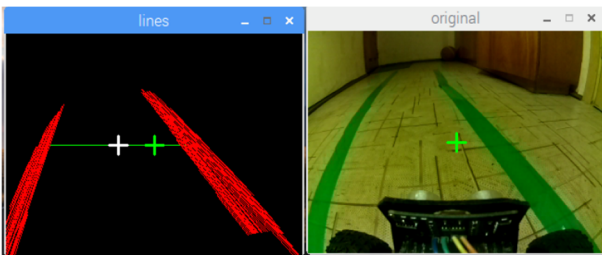
of the wheelbase.

The distance between green and white crosses determines the difference in camera and wheel orientation that must be compensated on the course. The program timer helps to synchronize the main program loop with Arduino response time. To obtain the position of 4WD platform on the path, MPU-6050 accelerometer module is used to get acceleration data in X-Y horizontal plan by the following methods [9-11].

The experimental results of processing the input stream of workspace are shown on Figure 4 and Figure 5.



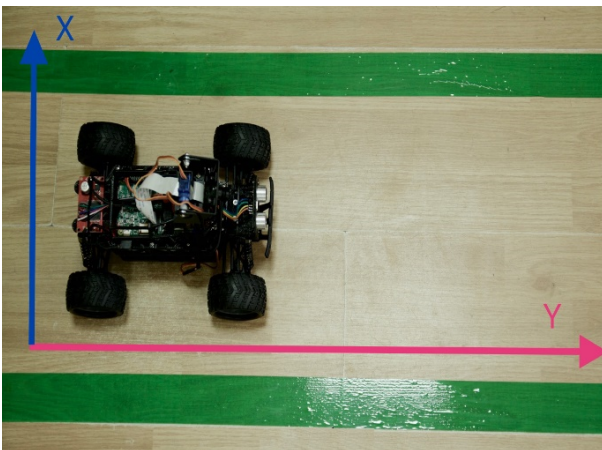
**Fig. 4.** The platform shifts to the left.



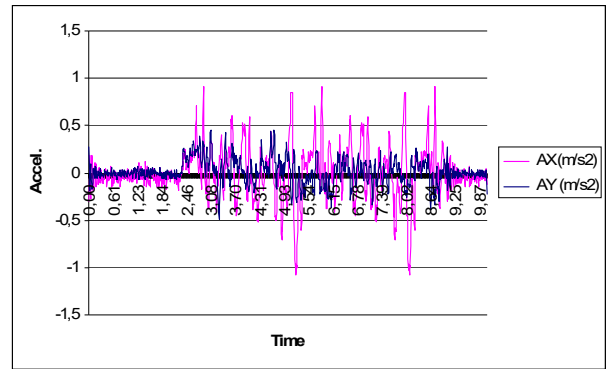
**Fig. 5.** The platform shifts to the right.

The source datasheet contains discrete time and acceleration along X and Y axes. After double integration it is possible to calculate velocity and relative position of wheeled platform. The result of steering control towards linear path, obtained with MPU-6050 accelerometer module for X-Y plan, is shown on Figure 6.

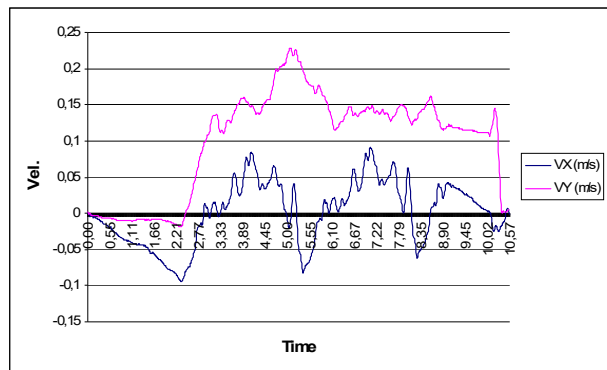
Reflected light near the starting position distorts the right guide line and steering control shifts to the right. At this case, the maximum deviation along the X axis reaches 12 cm (4WD platform does not cross the track border). Measured with the accelerometer relative distance is 107 cm which is close to planned 100 cm taking into account instrumental and computational errors.



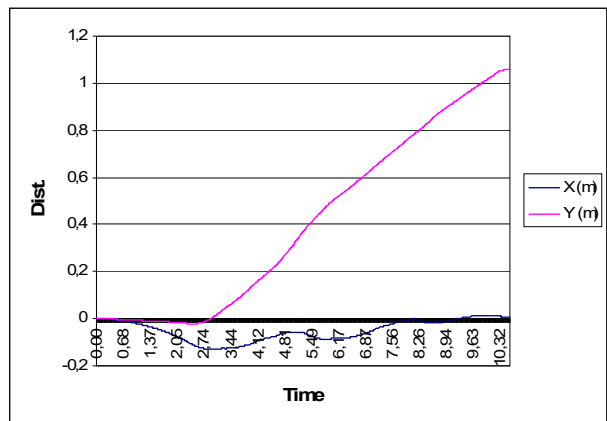
a)



b)



c)



d)

**Fig. 6.** Experimental results of tracking the linear path: a) Initial position of 4WD platform; b) auxiliary along X and Y axis; c) velocity along X and Y axis; d) path deviation along X and Y axis.

## 6 Discussion of research results

The AGV recognition system, built on a computer vision, gives more flexible approach to planning and maneuvering. The reliability of its use depends on the possibility of an error-free evaluation of the working area and consideration of possible situations in the control algorithms. Using a computer vision system allows to evaluate a large number of workspace options for navigation.

The operation of vision sensor is influenced by the various environmental conditions on which the result of recognition depends. One of the main problems of recognition is the uniformity of illumination and the

presence of reflecting surfaces. Even in artificial lighting conditions, the brightness level changes as a result of the appearance and disappearance of objects with different reflecting and light-absorbing properties. Extra artificial route marking, such as colored guide lines and markers, greatly simplifies the navigation task, as described in the study. The proposed approach, calculations and experimental studies of small 4WD platform show the possibility of fast navigation along the guide lines.

The limitations inherent in this study are the reflection of light leads to a mistake in steering the platform. However, as shown by the results of experimental part, the system still remains robust along the marked path, and the response of the control system is less than 0.2 seconds.

The prospect of further research is the realization of the maneuvering task around obstacles of complex geometry using object tracking by a camera with stabilization on the road using an accelerometer, as well as servicing many routes with visual marking.

## 7 Conclusion

1. The proposed 4WD platform provides autonomous movement and maneuvering using the steering system on a flat surface. The maximum radius of front wheels steering was 9 cm at a maximum speed of 0.8 m / s, which characterizes the high dynamics and maneuverability of navigating using guide lines.
2. The hybrid control system implements a terminal safety connection between the Raspberry PI and the Arduino Nano, allows extreme braking beyond the main control based on computer vision.
3. Implemented mathematical models and algorithmic-software implementation are intended to simplify the processing of data in the navigation direction along the guide lines. Estimating the lines perspective rate by computer vision allows additional control the speed on the way.
4. During experimental studies of autonomous driving stability on the road, the influence of uneven illumination and glare surfaces of the guide lines was estimated.

## References

1. Market Estimates & Trend Analysis. Automated Guided Vehicles Market (Grand New Research Report, 2016), <https://www.grandviewresearch.com/industry-analysis/automated-guided-vehicle-agv-market>. Accessed 22 Aug 2018
2. A.C. Pereira, F. Romero, A review of the meanings and the implications of the Industry 4.0 concept. *Procedia Manuf.* **13**, 1206–1214 (2017). doi:10.1016/j.promfg.2017.09.032
3. E<sup>n</sup>sor – Egemin Navigation System On Robot. DEMATIC (2018), <http://egeminusa.com/automated-guided-vehicles/software/ensor/>. Accessed 14 Feb 2019
4. N. Amin, M. Borschbach, Quality of obstacle distance measurement using Ultrasonic sensor and precision of two Computer Vision-based obstacle

- detection approaches. *IC-SSS* 1-6 (2015). doi:10.1109/SMARTSENS.2015.7873595.
5. M. Martínez, J. Martínez, J. Morales, Motion Detection from Mobile Robots with Fuzzy Threshold Selection in Consecutive 2D Laser Scans. *Electronics* **4**(1), 82–93 (2015). doi:10.3390/electronics4010082
6. D. Teso-Fz-Betoño, E. Zulueta, U. Fernandez-Gamiz, I. Aramendia, I. Uriarte, A Free Navigation of an AGV to a Non-Static Target with Obstacle Avoidance. *Electronics* **8**(2), 159 (2019). doi:10.3390/electronics8020159.
7. H. Tang, S. Shi, P. Huang, D. Wang, J. Zhou, PID Control of Magnetic Navigation Differential AGV Trajectory. *DEStech Transactions on Engineering and Technology Research*, 500–506 (2017). doi:10.12783/dtetr/apop2017/18774
8. A. Al-Mayyahi, W. Wang, P. Birch, Adaptive Neuro-Fuzzy Technique for Autonomous Ground Vehicle Navigation. *Robotics* **6**, 349–370 (2014). doi:10.3390/robotics3040349.
9. A.V. Koval', Simulation of gravimetric measurements by gyroscopic integrator of linear accelerations. *Gyroscopy and Navigation* **6**(1), 344–347 (2015). doi:10.1134/S2075108715040070
10. O. Bezvesilna, M. Kamiński, Gravimeters of aviation gravimetric system: Classification, comparative analysis, prospects. *Advances in Intelligent Systems and Computing* **550**, 496–504 (2017). doi:10.1007/978-3-319-54042-9\_48
11. O. Bezvesilna, A. Tkachuk, L. Chepyuk, S. Nechai, T. Khylichenko, Introducing the principle of constructing an aviation gravimetric system with any type of gravimeter. *Eastern-European Journal of Enterprise Technologies* **7**(1), 45–56 (2017). doi:10.15587/1729-4061.2017.92941
12. A. Rosebrock, *Practical Python and OpenCV: An Introductory, Example Driven Guide to Image Processing and Computer Vision*, 3rd edn. (PyImageSearch, 2016)
13. CarND Project 1: Lane Lines Detection – A Complete Pipeline (Toward Data Science, 2017), <https://medium.com/towards-data-science/carnd-project-1-lane-lines-detection-a-complete-pipeline-6b815037d02c>. Accessed 29 May 2019
14. G. Evensen, *Data assimilation: the ensemble Kalman filter*, 2nd edn. (Springer, Bergen, 2009)
15. PyCharm (JetBrains, 2000), <https://www.jetbrains.com/pycharm/>. Accessed 29 May 2019