Overhead Vision System for Testing Swarms and Groups of Wheeled Robots

Submitted: 16th January 2019; accepted: 16th May 2019

Jakub Wiech, Zenon Hendzel

DOI: 10.14313/JAMRIS/2-2019/14

Abstract:

The paper describes how to use ArUco markers to determine the position and orientation of wheeled robots in 3D space. It is preceded by a general description of the testbed and a detailed description of the marker detection algorithm along with the camera calibration using the ChaArUco markers. The camera has been described and calibrated using the pinhole camera model, taking into account distortion of the lens. The second part of the article describes the wheeled robots with their mechanical construction.

Keywords: *swarm robotics, vision system, wheeled robots*

1. Introduction

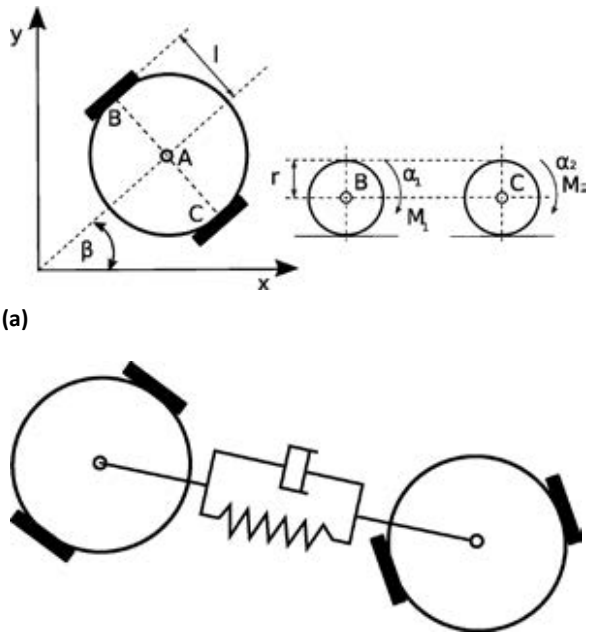
In mobile robotics one of the biggest challenges is determination of the robot's position and orientation in space. It can be achieved locally using on board sensors or globally using a external positioning system. In case of algorithm testing of groups and swarms of robots the best choice is a external positioning system. Depending on the requirements the positioning systems are either based on passive or active markers. The most precise as well as the most expensive systems are multicamera motion capture systems, for example: Vicon Vantage V5, OptiTrack Prime 17W or PhaseSpace X2E Impulse. These system are characterized by real time tracking with low latency for 6 DoF tracking of ground and aerial robots. Other cheaper ways of determining position and orientation are systems based on fiducial binary markers (Aruco, ARTag, RUNE-Tag) which are commonly used in virtual reality applications [2,4,11]. This approach is less precise but cheaper than the motion capture system.

In the case of robotic swarms, the local interaction between robots is important, i.e. robots locate and communicate, for example, only with their nearest neighbors. This is a necessary requirement to be able to distinguish a swarm from a multi-robotic system [1]. This requirement exist practically due to the limited range of sensors and communication modules, which is associated with equipping all robots with the right set of sensors. In order to verify the control algorithm, it is possible to replace the robot sensors with limited information on the nearest neighbors and obstacles provided by the external vision system. The information refers to all obstacles and robots that are at a distance R simulating the maximum range of the robot's sensors.

The paper presents a description of a vision system for determination of robots position and orientation in space and testing the control algorithms with example of leader following. For performance evaluation of proposed vision system the paper ends with four experiments and discussion of the results.

2. Robots

Objects used in experimental research are nonholonomic two-wheeled robots. Due to the targeted use of robots in multi-robot systems, robot groups or swarms, the small size of the robot (in this case 160 mm in diameter) is important, as well as a simple mathematical model describing the dynamics of the robot.



(a)

(b)

Fig. 1. Wheeled robot: a) – schematics, b) – virtual connection of robots

Where in (fig. 1), $\alpha = [\alpha_1, \alpha_2]^T$ are wheels rotation angles and generalized coordinates, $M = [M_1, M_2]^T$ are driving torques. The two-wheeled robot dynamics model is described by the matrix equations in papers [6] and [7].

2.1. Control Algorithm

The robot control algorithm is a follow-up control, the leader and the follower are following the desired trajectory, the important difference is that the tra-

jectory of the robot following the leader is generated online based on their relative position, the control scheme diagram is shown on (fig. 2). The follower maintains the desired distance from the leader acting as if it were connected by a virtual damped spring (fig. 1b). Spring parameters are selected experimentally. Robots are controlled by a PD controller.

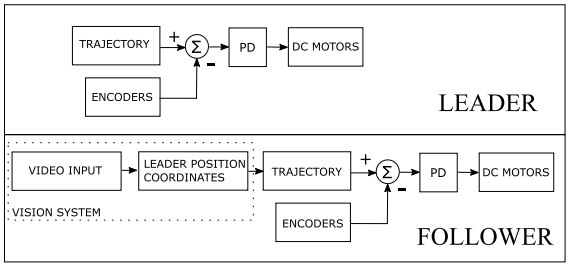


Fig. 2. Control scheme diagram

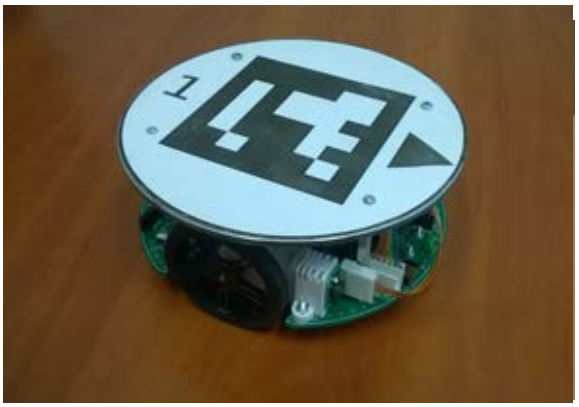
2.2. Robot Design

The mobile robots used in experimental research are symmetric two-wheeled robots with two ball casters (fig. 3a). The robot is equipped with a WiFi module for communication with an external vision system that transmits the distance value between the geometric center of the robot and the centers of its nearest neighboring robots along with their angular position relative to the i-th robot. The robot PCB is also its frame to which the motors and robot support wheels are mounted (fig. 3b).

The robot is controlled by the popular 8 bit Atmega2560 microcontroller, which allows programming in several languages. Assembler, C, Arduino language and also by using custom libraries, it's possible to program the robot directly through Arduino add-on in MathWorks Matlab / Simulink package . The last solution significantly speeds up the process of design and verification of the control algorithm. With a large number of robots, it is important to easily charge and reprogram the robots, it can be achieved by adding an additional wireless communication module and adjusting the robots to use the docking station or contact charging.

3. Robotic Testbed

The robotic testbed is used for experimental research and testing of control algorithms of swarm or a group of wheeled robots. It consists primarily of two systems, a vision system and a wireless communication system (fig. 4). The vision system is intended to determine the location of a given robot in space and its position relative to other robots. The camera of the vision system is located above the surface on which the robot movement is tested at a distance enabling obtaining the desired accuracy of the position and orientation of the robots in the group. The wireless communication system is based on the WiFi wireless communication standard and enables the reading of measurement data from the robot, such as the angular velocity of the driving wheels. Vision system and wireless communication system are controlled from a



(a)

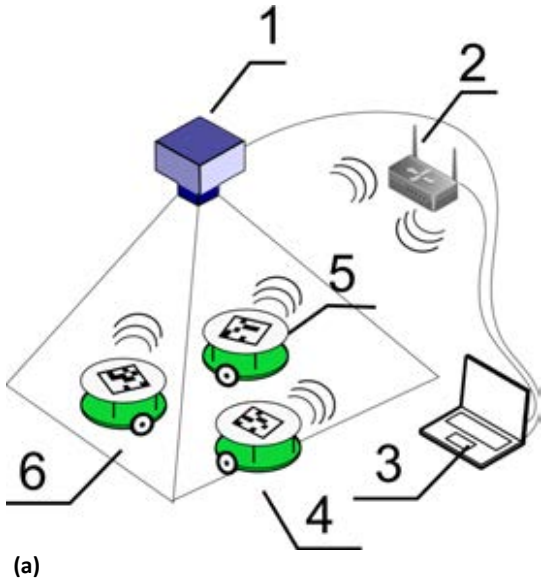
(b)

Fig. 3. Wheeled robot: a) – with marker b) – without marker

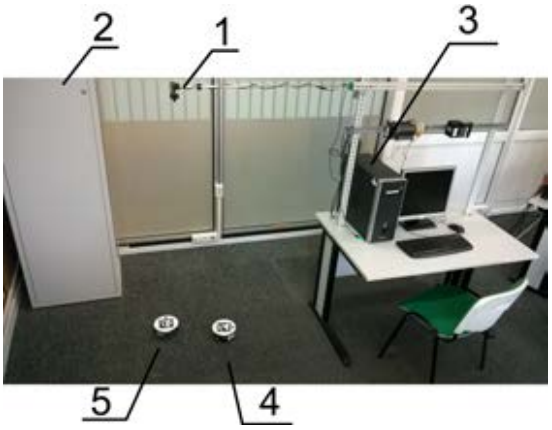
computer application that allows to view the camera image, send commands as well as receive and display data from mobile robots. The tested robots are two-wheeled mobile robots. Two-wheeled robots are often used to verify the control algorithms of groups and swarms of wheeled robots due to their simplicity and low construction cost. In the case of swarms, this is a big advantage considering the requirements of the swarm existence, i.e. large numbers of robots and possibly the simplest mechanical structure of the robot, which translates into low costs of series production.

3.1. The Vision System Algorithm

Identification and determination of the robot position and orientation in space is possible by using ArUco markers. ArUco fiducial marker are square binary coded tags with identification number, with the control bits. Image processing algorithm based on four corners of the marker finds the coordinates x, y, from the center of the marker. The robot binary identification number is compared to the previously prepared database of robots' ID's. Based on the identification number and marker orientation, the algorithm determines the orientation of all robots in the field of view of the camera. Information about the orientation and location of the robot is recorded and depending on the used control algorithm, it can be sent to any robot. Another type of ArUco markers is the ChArUco checkerboard. The ChArUco marker is a chessbo-



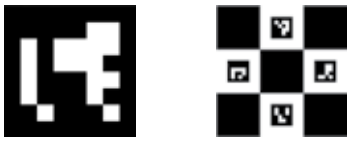
(a)



(b)

Fig. 4. Research testbed: 1- camera, 2- router, 3-computer, 4,5,6- mobile robots: a) – schematic diagram b) – research testbed

ard with ArUco markers inside the white chessboard boxes (fig. 5). It is usually used to calibrate the camera, allowing the calibration of partially covered markers, which is not possible with an ordinary black and white chessboard.



(a) (b)

Fig. 5. Marker examples: a) – ArUco marker b) – ChArUco chessboard

3.2. Camera Calibration

Before using the camera, it should be calibrated. Calibration is carried out once, in case of using a new camera or after changing its construction, for example after changing the lens. For camera calibration, black and white checkers or ChArUco markers are used in various angular positions with respect to the camera (fig. 6). Knowing a few or a dozen items on the chess-

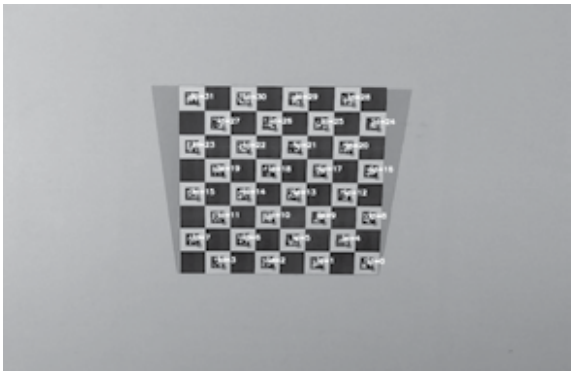


Fig. 6. Calibration process

board using the method described in [14] we are able to determine the internal parameters of the camera. Camera calibration process is in other words the determination of the matrix of internal camera parameters, lens distortion parameters and the rotation and translation matrix describing the transformations between the global coordinate system and the local camera coordinate system. The matrix of internal parameters together with image distortion parameters are used for projection of x, y, z coordinates from the camera described in the plane of the camera image. The internal parameters of the camera are of form

$$A = \begin{bmatrix} f_1 & \gamma & u_0 \\ 0 & f_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where, f_1, f_2 - focal lengths in pixels, u_0, v_0 - coordinates of the image center, γ - axis scale factor.
The rotation and translation matrix is written in a combined form, i.e.

$$[R|t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}, \tag{2}$$

where, r_{ij} - element of rotation matrix, t_1 - element of translation matrix.
The matrices of internal camera parameters as well as rotation and translation are used in the simplest camera model, i.e. the pinhole camera model. A pinhole camera is a camera without a lens with a small aperture in the shape of a small hole. The light passing through the hole casts the inverted image of the object on the camera sensor. In such a simplified model, the relationship between coordinates of a point in a three-dimensional global system and the coordinates of the projection of the point on the camera image are expressed by the formula

$$s \begin{bmatrix} U_i \\ V_i \\ 1 \end{bmatrix} = A[R|t] \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}, \tag{3}$$

$$A[R|t] = \begin{bmatrix} f_1 & \gamma & u_0 \\ 0 & f_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$

where, U_i, V_i - coordinates of the point on the image plane in the camera reference system according to the pinhole camera model, s - scale factor.
If the camera is equipped with a lens, the pinhole camera model is insufficient to properly transform the coordinates of the image. The presence of the lens introduces image distortion and thus complicates the description of the camera model. We can distinguish two types of distortion. Radial distortion causing image distortion symmetric to the lens radius, examples are barrel and pincushion distortion. The second type of distortion is tangential distortion resulting from non-parallelism of the lens with the camera optical sensor (fig. 7).

Fig. 7. Examples of distortion of the image: a) – radial pincushion b) - radial barrel c) - tangential

The correct camera model for calibration is derived by combining the pinhole camera model with the correction of radial and tangential distortions. The model describes a matrix equation

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} D_u s_u \left(\tilde{u_i} + \delta u_i^{(r)} + \delta u_i^{(t)} \right) \\ D_v \left(\tilde{v_i} + \delta v_i^{(r)} + \delta v_i^{(t)} \right) \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}, \quad (4)$$

where, u_i, v_i - coordinates of the point on the image plane in the camera's reference frame after taking into account distortion of the image, $\tilde{u_i}, \tilde{v_i}$ - projection of coordinates of a point on the image plane, $\delta u_i^{(r)}, \delta v_i^{(r)}$ - coordinates of the point in the image plane shifted radially, $\delta u_i^{(t)}, \delta v_i^{(t)}$ - coordinates of a point on the image plane shifted due to tangential distortion, D_u, D_v - conversion factors of millimeters to pixels, s_u - scale factor.
The coordinates of a point $P_i\,(x_i, y_i, z_i)$ on the image plane corresponds to the relationship resulting from the pinhole camera model

$$\begin{bmatrix} \tilde{u_i} \\ \tilde{v_i} \end{bmatrix} = \frac{f}{z_i} \begin{bmatrix} x_i \\ y_i \end{bmatrix}. \quad (5)$$

Radial distortion $\delta u_i^{(r)}, \delta v_i^{(r)}$ and tangential distortion $\delta u_i^{(t)}, \delta v_i^{(t)}$ are expressed by equations

$$\begin{bmatrix} \delta u_i^{(r)} \\ \delta v_i^{(r)} \end{bmatrix} = \begin{bmatrix} \tilde{u_i} \left(1 + k_1 r_i^2 + k_2 r_i^4 + k_3 r_i^6 \right) \\ \tilde{v_i} \left(1 + k_1 r_i^2 + k_2 r_i^4 + k_3 r_i^6 \right) \end{bmatrix}, \quad (6)$$

$$\begin{bmatrix} \delta u_i^{(t)} \\ \delta v_i^{(t)} \end{bmatrix} = \begin{bmatrix} 2p_1 \tilde{u_i} \tilde{v_i} + p_2 \left(r_i^2 + 2\tilde{u_i}^2 \right) \\ 2p_2 \tilde{u_i} \tilde{v_i} + p_1 \left(r_i^2 + 2\tilde{v_i}^2 \right) \end{bmatrix}, \quad (7)$$

where, k_1, k_2, k_2 - radial distortion coefficients, p_1, p_2 - tangential distortion coefficients, $r_i = \sqrt{\tilde{u_i}^2 + \tilde{v_i}^2}$.

The camera used on the research testbed is a camera with a resolution of 1920x1080p, 120 FPS with an OmniVision OV4689 optical sensor . After performing the calibration algorithm, the following camera parameters were obtained

$$A = \begin{bmatrix} 1262 & 0 & 371 \\ 0 & 1284 & 637 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} -0.26 \pm 0.03 \\ -1.36 \pm 0.03 \\ 2.86 \pm 0.03 \end{bmatrix}, \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 0.02 \pm 0.01 \\ 0.05 \pm 0.01 \end{bmatrix} \quad (8)$$

3.3. Marker Detection Algorithm

Detection of markers is carried out on the basis of an algorithm developed in publication [5]. The process of ArUco markers detection and binary code extraction is shown in (fig. 8)

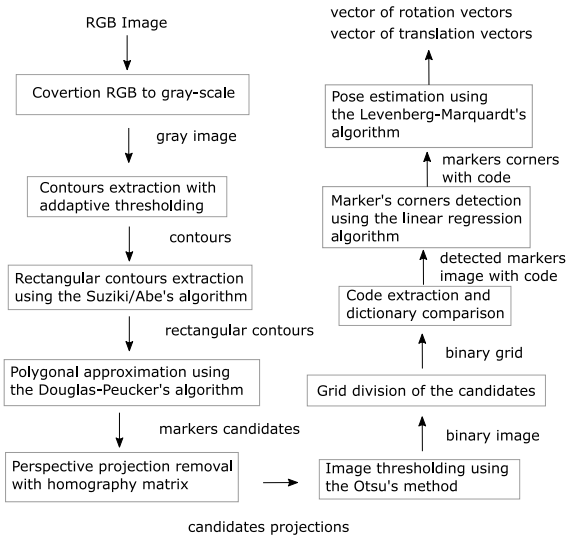


Fig. 8. Marker detection algorithm

Marker detection algorithm can be divided into 5 stages:

1) Image segmentation - extraction of the most distinctive contours from camera image presented in shades of gray using local adaptive thresholding .
2) Extraction of contours and image filtration - obtained image with highlighted contours is subjected to the algorithm in from [3], giving a set of contours from which emerge rectangular contours resembling ArUco markers .
3) Extract binary code markers - the first step is to change from perspective projection for orthogonal projection using a homographic matrix and performing thresholding from [9]. The resulting binary image is divided by a rectangular grid where each cell is assigned a value of 1 or 0 depending on the color of the cell.
4) Marker identification and error correction - Identification numbers are determined based on the binary value inside the marker border for all 4 possible marker rotations. The obtained values are compared with the dictionary of possible marker identification numbers.

5) Corners position determination and pose estimation of the marker - knowledge of the position of the marker's corners is required to determine the marker pose. In the case of ArUco, the corners of the markers are determined by the linear regression algorithm of the pixels on the edge of the marker to determine their intersection.

Aruco library available in OpenCV repository in Python language allows for markers detection using the command $cv2.aruco.detectMarkers$. For marker's pose estimation the command $cv2.aruco.estimatePoseSingleMarkers$ can be used.

4. Experiments

To determine the marker localization error a radial positioning error map was experimentally obtained. The robot arena was filled with 360 markers spaced $68mm$ from each other forming a grid (black dots). The e_x and e_y positioning errors of each marker were used to calculate the radial error $e_r = \sqrt{e_x^2 + e_y^2}$ which is shown in color on the error map (fig. 9).

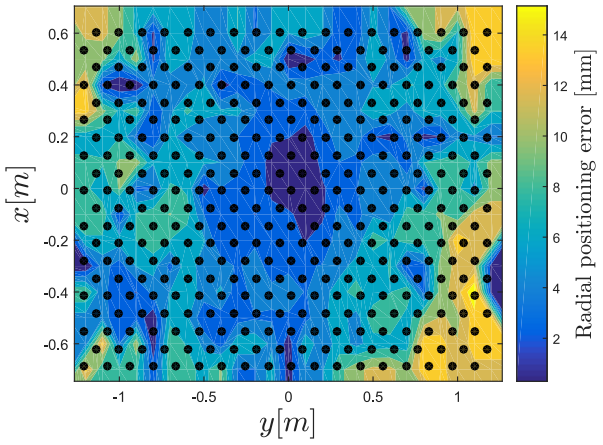


Fig. 9. Vision system error map

As it is shown on the error map, the radial error increases with the distance from the center point (0,0) and its highest value reaches 15 mm in the corners of the camera field vision.

To evaluate the usefulness of the proposed vision system in testing swarms and groups of robots a relation between the camera FPS and number of robots was determined (fig. 10).

After necessary calculations for marker detection and pose estimation the camera fps drops form 120 to $22fps$ with addition of new robots (fig. 10). In case of 13 markers fps drops to $60fps$, for 24 markers drops below $30fps$. The experiment shows that the vision system is well suited for groups and swarms of robots up to 24 robots. For $30fps$ the minimum step size in robot control algorithm is $0.03s$, bigger step size would result in unacceptable numerical errors in the control algorithm.

For performance evaluation of the proposed vision

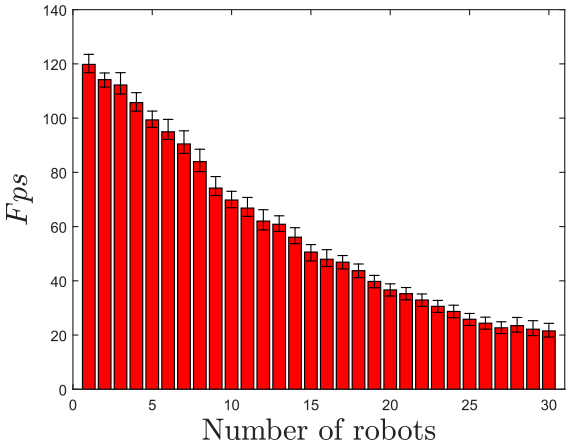


Fig. 10. Change of fps with increase of numbers of robots

system in context of robotics two experiments were performed. The vision system determines the position and orientation of markers. In the first experiment a marker is placed on a programmable slider moving the set distance. In the second experiment the vision system is used in example of leader following. In both cases the values of marker's position and orientation are compared with values from encoders.

4.1. Programmable Slider

The Aruco Marker is placed on the moving programmable slider traversing the desired distance with given velocity slope. The marker is rotating along its axis with set angle and angular velocity. The slider is being moved by a toothed belt connected to a stepper motor with an encoder. The marker is mounted on the servo-motor shaft located on the moving cart (fig. 11) [12].

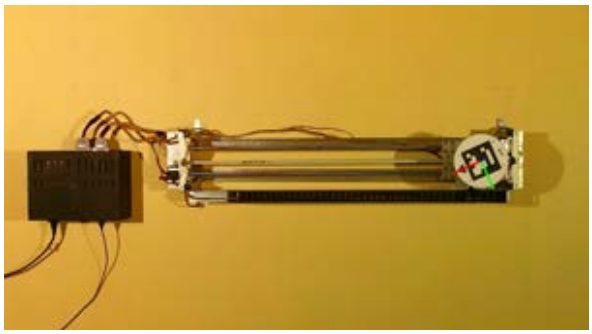
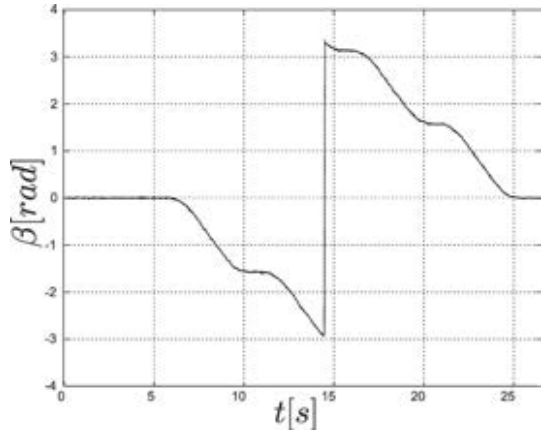
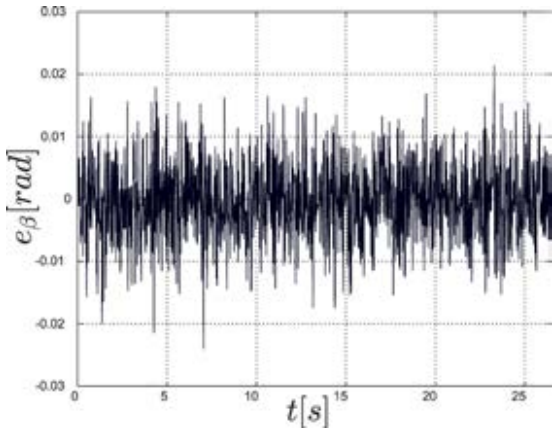


Fig. 11. The slider

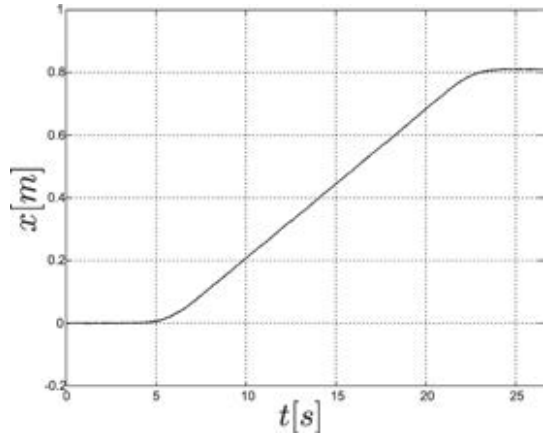
The slider was positioned along the x axis of the camera. The sliders set distance was $0.81m$ from the first limit switch to the next and the marker set rotation angle was $4\frac{\pi}{2}$. The resulted sliders traverse distance and marker rotation angle are depicted in (fig. 12c) and (fig. 12a). The maximum positioning error in x axis was $2mm$ (fig. 12d) whereas the maximum orientation angle error was $0.02rad$ (fig. 12b).



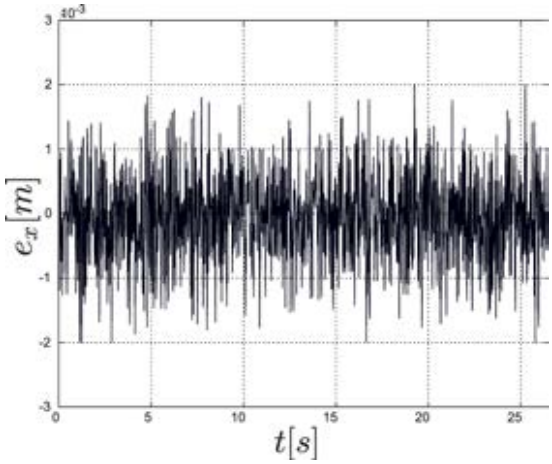
(a)



(b)



(c)

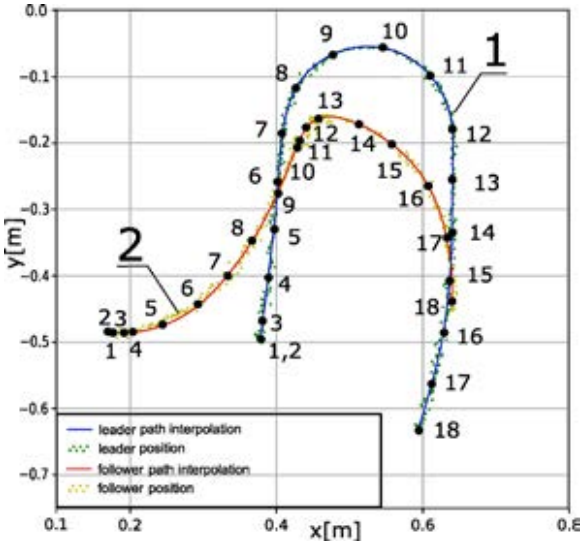


(d)

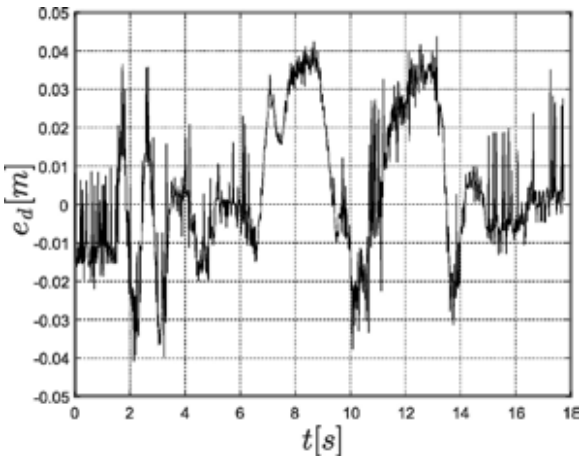
Fig. 12. Plots of the marker orientation and position values from vision system: a) – the angle of orientation β of the marker b) – the angle of orientation error based on odometry and camera data c) – marker position d) – marker position error based on odometry and camera data

4.2. Following the Leader

The leader marked as 1 follows the set trajectory while the following robot 2 has to follow the leader keeping the distance $20cm$ from the center of the leader.



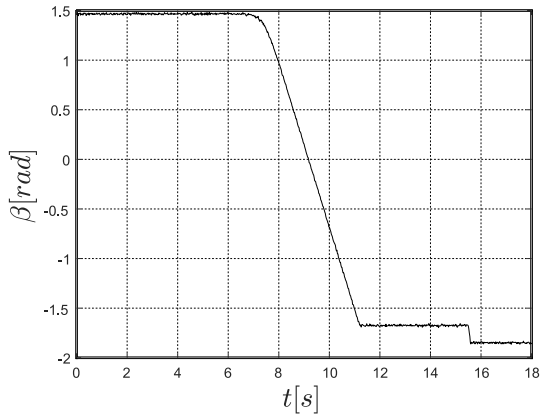
(a)



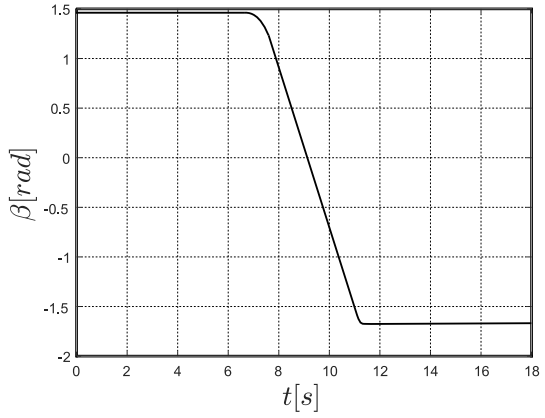
(b)

Fig. 13. Leader following: a)- paths of leader's and follower's geometric centers (with specified positions every 1s) ,b) – the distance error of maintaining desired distance between robots from the vision system

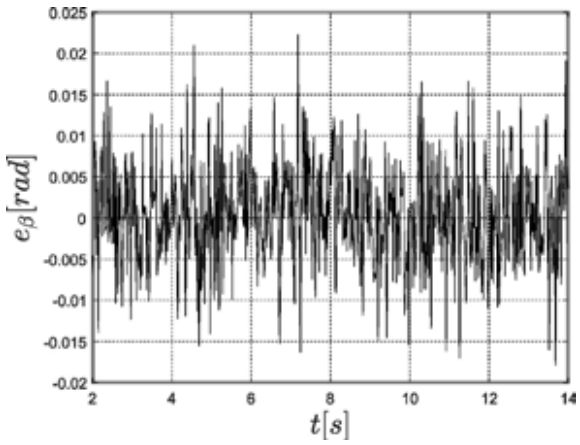
The paths were obtained by approximating the measuring points from the vision system. The average difference in distance in individual time moments marked on the movement path, (fig. 13) is $20cm$. The maximum distance error is $0.042m$, while the average error is $0.041m$. To determine the position and orientation of the camera the odometry was used. The



(a)



(b)



(c)

Fig. 14. Plots of the β angle of the leader frame and robot frame orientation error: a) – the angle of orientation β of the leader frame determined by the vision system b) – the angle of orientation β of the leader frame determined by the odometry c) – the difference of the leader's orientation angle based on odometry data and camera from 2s to 14s

change of distance error between robots is shown in (fig. 13b), while the difference in the angle β of the leader frame on the basis of data from odometry and camera is shown in (fig. 14c). Based on the chart of the slope of the leader's frame (fig. 14b), it can be noticed that in $15.5s$ there was a wheels slip, which is also visible on the leader's path in (fig. 13a). The leader had to move on a straight line, make one loop and continue to move along the straight line. The shape of the obtained trajectory is affected by the blurring of the image resulting from the movement of robots and the initial placement of the robots. Due to the low speed, we assume that the image of the markers is sharp. In order to correctly assess the error of determining the position and orientation of the robots, it is necessary to compare the results with an additional video system with known camera parameters. To determine the estimate values of the position and orientation error, we compare the measurements with the known position and orientation of the marker. The uncertainty of determining the position of the markers is: $\Delta x = 2mm$, $\Delta y = 3mm$ while orientation $\Delta\beta = 0.02rad$.

5. Results Discussion

To evaluate whether the results we gathered are correct we will compare our results with known publications on using the Aruco Markers with position and orientation accuracy estimation.

According to our research the radial error increases with the distance from the center of the camera field vision (point (0,0)). Similar results were obtained in [8] where Aruco Markers were tested in virtual experiments. The radial error is dependent with the distance between a marker and the camera. " The further a marker is from the camera, the bigger the error in the 3D estimations. Under $4m$ the distance error keep below $5cm$". The same is true in our experiment. The biggest difference is in case of the orientation error. The Authors have shown that the orientation error is below $0.02°$ under $4m$ of distance camera-marker. In our case the error is $0.02rad$. The discrepancy may be the result of the "inaccuracy of method used to estimate the real true pose" as the Authors concluded.

In publication [13] the Authors used Aruco markers for visual SLAM and concluded that the mean positioning error is less than 2% calculated from the distance camera-marker. In the distance equal $2.5m$ the positioning error is less than $5cm$ which is the same as in previous publication.

It is worth adding that when the marker is directly below the camera the positioning error drops below $1mm$ (fig. 9). In publication [10] the Authors have shown that it is possible to achieve positioning accuracy less than $0.075mm$ along X and Y axis and $0.3mm$ along Z axis. Which explains high accuracy in our experiments near point (0,0).

6. Summary

The above article shows how thanks to the previously calibrated camera, the vision system allows registering the position and orientation of up to 24 mobile robots in space with 30 fps or 13 robots with 60 fps. The ArUco marker, located in the geometric center of the robot, gives the opportunity to track the actual trajectory of the robot. The accuracy of the position

measurement depends on the resolution and accuracy of the camera calibration as well as the distance from center of the camera field vision. The longer the distance the higher the positioning error. Simplicity of robot construction, ease of programming and availability of programming libraries such as OpenCV and ArUco enable easy implementation and verification of robot control algorithms.

AUTHORS

Jakub Wiech* – Rzeszow University of Technology, Faculty of Mechanical Engineering and Aeronautics , e-mail: j.wiech@prz.edu.pl.
Zenon Hendzel – Rzeszow University of Technology, Faculty of Mechanical Engineering and Aeronautics , e-mail: zenhen@prz.edu.pl.
*Corresponding author

REFERENCES

[1] G. Beni, "From Swarm Intelligence to Swarm Robotics". In: E. Şahin and W. M. Spears, eds., *Swarm Robotics*, 2005, 1–9, 10.1007/978-3-540-30552-1_1.

[2] F. Bergamasco, A. Albarelli, E. Rodolà, and A. Torsello, "RUNE-Tag: A high accuracy fiducial marker with strong occlusion resilience". In: *CVPR 2011*, 2011, 113–120, 10.1109/CVPR.2011.5995544.

[3] D. H. Douglas and T. K. Peucker, "Algorithms for The Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature", *Cartographica: Int. J. Geogr. Inf. Geovis.*, vol. 10, no. 2, 1973, 112–122, 10.3138/FM57-6770-U75U-7727.

[4] M. Fiala, "ARTag, a fiducial marker system using digital techniques". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, 2005, 590–596, 10.1109/CVPR.2005.74.

[5] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion", *Pattern Recognition*, vol. 47, no. 6, 2014, 2280–2292, 10.1016/j.patcog.2014.01.005.

[6] J. Giergiel and W. Żylski, "Description of motion of a mobile robot by Maggie's equations", *Journal of Theoretical and Applied Mechanics*, vol. 43, no. 3, 2005, 511–521.

[7] M. Giergiel, Z. Hendzel, and W. Żylski, *Modelowanie i sterowanie mobilnych robotów kołowych*, Wydawnictwo Naukowe PWN: Warszawa, 2013.

[8] A. Lopez-Ceron and J. M. Canas, "Accuracy Analysis of Marker-Based 3D Visual Localization". In: *XXXVII Jornadas de Automatica Workshop*, 2016.

[9] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, 1979, 62–66, 10.1109/TSMC.1979.4310076.

[10] D. C. Popescu, M. O. Cernaianu, P. Ghenuche, and I. Dumitrache, "An assessment on the accuracy of high precision 3D positioning using planar fiducial markers". In: *21st International Conference on System Theory, Control and Computing (ICSTCC)*, 2017, 471–476, 10.1109/ICSTCC.2017.8107079.

[11] P. C. Santos, A. Stork, A. Buaes, C. E. Pereira, and J. Jorge, "A real-time low-cost marker-based multiple camera tracking solution for virtual reality applications", *Journal of Real-Time Image Processing*, vol. 5, no. 2, 2010, 121–128, 10.1007/s11554-009-0138-9.

[12] J. S. Tutak and J. Wiech, "Horizontal Automated Storage and Retrieval System", *Advances in Science and Technology Research Journal*, vol. 11, no. 1, 2017, 82–95, 10.12913/22998624/68470.

[13] R. S. Xavier, B. M. F. d. Silva, and L. M. G. Gonzalves, "Accuracy Analysis of Augmented Reality Markers for Visual Mapping and Localization". In: *2017 Workshop of Computer Vision (WVC)*, 2017, 73–77, 10.1109/WVC.2017.00020.

[14] Z. Zhang, "A flexible new technique for camera calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, 2000, 1330–1334, 10.1109/34.888718.