Control System of a Semi-Autonomous Mobile Robot ⋆

Pavel Škrabánek * **Pavel Vodička** * **Sule Yildirim-Yayilgan** **

* *Faculty of Electrical Engineering and Informatics, University of Pardubice, Studentská 95, Pardubice, 532 10, Czech Republic (e-mail: pavel.skrabanek@upce.cz)*
** *Faculty of Computer Science and Media Technology, Norwegian University of Science and Technology, postboks 191, NO-2802 Gjøvik, Norway (e-mail: sule.yildirim@ntnu.no)*

Abstract: In this paper, a control system, which has been designed for a semi-autonomous mobile robot, is presented. The robot has been developed as a part of a teaching aid. The teaching aid is aimed at practice of a path-planning theory. The main purpose of the robot within the teaching aid is physical execution of a path-plan scheduled by a high-level control system. The execution of path-plans is required to be accomplished by the robot autonomously, hence the 'semi-autonomous' appellation. The robot is based on an Arduino UNO microcontroller board. The robot acquires information about his workspace via reflectance sensors, encoders, and a magnetometer sensor. Since all these sensors provide only very limited information about the workspace, all the acquired data has to be used with utmost effectiveness. Thus, processing of the sensor data and multi-sensor integration is also considered in this paper.

Keywords: autonomous mobile robots, computer control system design, multisensor integration, path planning, data processing.

1. INTRODUCTION

Mobile robots are popular teaching aids within the education system. Many different types of mobile robots suitable for education have been introduced. They have been employed in different environments where diverse activities are required to be performed, e.g. line following (see Pakdaman and Sanaatiyan (2009)), collecting stuff, obstacle avoidance, or solving a maze (see Alves et al. (2011)).

Development of a teaching aid aimed at use of a mobile robot is a complex process. The process can be considerably simplified once a commercial solution is used, e.g. Cuéllar and Pegalajar (2014) used LEGO Mindstorms kit, or Harlan et al. (2001) and Rubenstein et al. (2014) employed products from a Swiss company called K-Team Corporation. Although many different commercial products are available on the market, it is difficult to find an inexpensive solution which perfectly fits a specific teaching purpose. Of course, as Mariška and Doležel (2014) showed, it is possible to use a virtual environment; nevertheless, students then lose the connection with reality.

A good experience with mobile robots while teaching path-planning techniques, presented by Alves et al. (2011), motivated us to enrich our courses on artificial intelligence with a practical exercise based on this experience. The lack of inexpensive solutions suitable for this purpose led us to develop a new teaching aid tailored to our requirements.

⋆ The work has been supported by the Funds of University of Pardubice, Czech Republic, grant No. SGS-2016-017.

This teaching aid consists of a maze, a camera system, a high-level control system, and a mobile robot. The goal of intended exercises, which will be realized by the teaching aid, is practicing of path-planning theory treated in a course 'Introduction to Artificial Intelligence 1'. The path-planning is supposed to be performed by the high-level control system while its execution is objective of the mobile robot. In this conception, autonomous execution of a path-plan is required by the mobile robot; hence the appellation 'semi-autonomous'. An autonomous execution of a path-plan is really not a simple task. Many different tasks have to be carried out by a control system of an autonomous or a semi-autonomous mobile robot. In this paper, description of a control system, which has been developed for the aforementioned mobile robot, is presented.

The rest of the paper is organized in following way. In section 2, the teaching aid is introduced and its components are detailed. In the same section, the intended use of the teaching aid is discussed in context of the path-plan execution. Physical execution of path-plans is conditional on sufficiently accurate knowledge of the environment by the robot. The robot gathers information about the environment via sensors. Considering this, the sensors of the mobile robot are explored in depth in section 3. Analysis of the control problem is considered in section 4. On the basis of the analysis, the control system has been divided into two levels. The higher level is presented in section 5 while the lower one is considered in section 6. The whole work is then summarized in section 7 where also advantages and disadvantages of the presented solution are discussed.

2. TEACHING AID

In this paper, a control system of the mobile robot, which has been developed as a part of the teaching aid, is presented. Since the design of the control system has been significantly influenced by an overall idea behind the teaching aid, the purpose of the teaching aid is explained in subsection 2.1. As has been already mentioned, the teaching aid consists of four physical parts: the *maze*, the *camera system*, the *mobile robot*, and the *high-level control system*. Except for the high-level control system, all physical parts of the teaching aid had to be considered within the development process.

In order to clarify the presented solution, all relevant parts of the teaching aid are detailed in this section. Specifically, the maze is presented in subsection 2.2; the camera system in subsection 2.3; and finally, construction of the mobile robot is described in subsection 2.4. Nonetheless, not only the physical parts of the teaching aid have affected the development process. The path-planning, which is analyzed in subsection 2.5, was equally important in the design of the control system.

2.1 Purpose of the Teaching Aid

As has been mentioned in the introduction, the teaching aid is aimed for practicing the path-planning theory. Within the exercises, students are supposed to make a *path-planning routine* according to a given assignment. The routine may find the shortest path between an actual position of the robot and a designated point in the maze. The target position is determined by a user.

In the presented teaching aid, the high-level control system collects all information which is necessary for the path-planning. Information about the current state of the robot, as well as about structure of the maze, is gathered via the camera system. The high-level control system also provides a user friendly environment where a user can simply determine a target position. However, no path-planning routine is integrated in this system. For the path-planning, routines developed by students are used. The high-level control system provides the acquired information in a suitable form to a currently employed path-planning routine. A *path-plan* provided by the routine is then physically executed by the robot.

2.2 Maze

The maze is a workspace of the robot. In order to offer a flexible solution allowing creation of different environments, a maze building kit has been developed. This kit has been designed as a modular system which consists of partitions of a fixed size, floor blocks, and posts. The dimensions of the partitions are $170 \times 170 \times 100$ mm and dimensions of the posts are $12 \times 12 \times 100$ mm. Here and further, dimensions of any object are written as length $\times$ width $\times$ height.

To each post, up to four partitions can be inserted into preformed tracks. The tracks are parallel to the longest edges of the posts and just one track is on each side of the post. Thus, the kit allows creation of different environments of rectangular layout where the partitions and the posts are obstacles from the perspective of the robot. Nonetheless, within the exercises, structures of mazes are limited by two requirements: the outer shape of a maze has to be rectangular, and a maze has to be a closed system (isolated from a surrounding world). For more information, see Škrabánek (2015).

In the context of this paper, surface coatings of partitions and posts are also very important, therefore they are detailed in this subsection, too. The partitions have been painted by a white color. A shade of the white is identical for all the partitions. Further, there are no significant irregularities on surfaces of the partitions. The posts are made from aluminum without any finishing. The posts can be also considered to be flat.

2.3 Camera System

The camera system is the only source of data about the real world, which is available to the high-level control system. The camera system consists of a stand arm and a camera. The camera is fixed on the stand arm and the whole camera system is placed such that the camera is above the maze. A procedure, which was presented by Škrabánek (2015), extracts information about the maze structure from captured images. The procedure provides this information in a form of a graph. However, as Škrabánek and Doležel (2014) showed, the camera system can be also used for an acquisition of information about a current state of the robot.

2.4 Mobile Robot

In the discussed teaching aid, a differential wheeled mobile robot is used. The robot has been developed at the University of Pardubice, Faculty of Electrical Engineering and Informatics, Department of Process Control. The robot is based on a microcontroller board named Arduino UNO, rev. 2. Communication with other electronic devices is possible either via USB cable or via Bluetooth module HC-05. Naturally, the Bluetooth module is used for communication with the high-level control system when the robot operates in the maze. The final version of the robot is shown in Fig. 1. As is obvious, the shape of the robot body is approximately a block of dimensions $110 \times 94 \times 83$ mm (without the Bluetooth module). A coordinate system relative to the robot body is depicted in the left down corner of the figure.

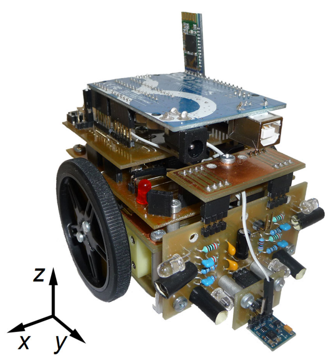

Fig. 1. Final version of the robot from the front view

The robot is driven by two quadruple high-current half-H drivers L293D. Speed of a wheel rotation can be determined on the basis of data provided by an incremental optical encoder, which is attached on the inner side of the wheel. Reading of the data is implemented using reflective object sensors QRD1114. The position of the robot, with respect to obstacles, can be estimated on the basis of data provided by two front and two side reflective sensors. The sensors are based on infrared emitters L-53F3C and phototransistors L-53P3BT. The robot is further equipped with one triple axis magnetometer sensor HMC5883L. Approximate placements of the drivers and the sensors, except for the encoders, are shown in Fig. 2. The coordinate system relative to the robot is stated at each view in this figure.

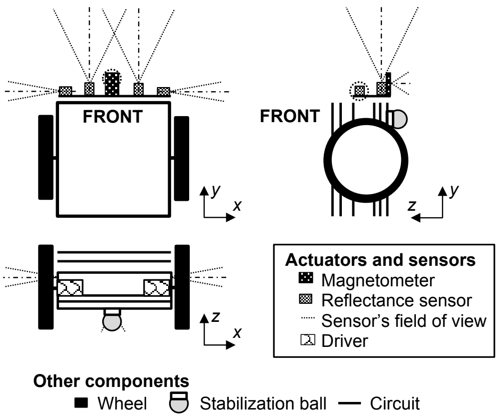

Fig. 2. Placement of sensors and drivers in the robot body

2.5 Path-Planning

Practicing of the path-planning theory is the primary purpose of the teaching aid. Namely, searching for a single pair shortest path in a maze is the task to be solved by students. The students are asked to program a path-planning algorithm in the form of a routine of three inputs and one output. The inputs are: information about a maze structure provided as an *adjacency matrix*; information about an *initial* (actual) *position* of the robot; and information about a *target position* of the robot specified by a user. The output of the routine is a path-plan expressed as a *sequence of vertices* the robot should pass through.

From the previous text, it follows that a path-plan is aimed to be formulized by a path-planning routine on the basis of a discrete model of the workspace. As was pointed out by Škrabánek (2015), *exact cell decomposition* is appropriate for discretization of the workspace shaped by a maze. Indeed, the dimensions of the robot, the standardized dimensions of the partitions, a flat floor, and the rectangular layout of the maze, directly encourage the forming of a 2D grid of square *cells* of size 184×184 mm over the maze. Borders of such created cells coincide with feasible positions of the partitions as is shown in Fig. 3. The numbers stated in the figure are order numbers of the cells.

From the perspective of the path-planning, such created cells are vertices v of a graph $G = (V, E, w)$, where V is a set of vertices v, E is set of edges e, and w are weights

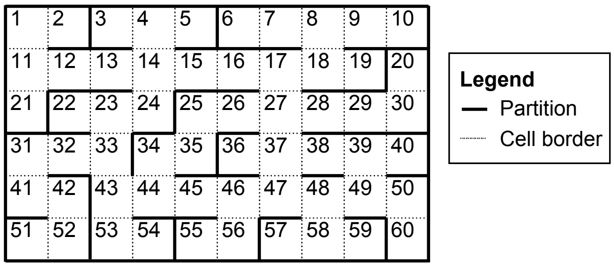

Fig. 3. Decomposition of a maze into the discrete representation

of the edges. In the case that no partition between two adjacent cells is presented, transition between these two cells is possible, and a corresponding edge e is included into the set E. Since the floor is flat, weights w are equal to 1 for $\forall e \in E$. For more information about this topic, see Škrabánek et al. (2015).

Within the conception of the teaching aid, information about a maze structure is provided to the high-level control system by the camera system in the form of an image. A transformation of this information into the discrete model is carried out by the high-level control system. Specifically, a routine introduced by Škrabánek (2015) is used. This routine provides the discrete representation of the maze directly as the adjacency matrix.

Naturally, the remaining two inputs of the path-planning routine might be expressed in a form which matches the used model. Thus, they are also given as vertices. The target position of the robot can be directly expressed by the user as a desired vertex. Somewhat more difficult is the position of the robot. In our case, mapping of a physical position of the robot into a nearest cell is used. To be more specific, distances between the geometrical center of the robot and geometrical centers of four nearest cells are assessed. In such a way, the robot's initial position is positively located to exactly one of the vertices $v \in V$.

3. SENSORS AND DATA PROCESSING

The mobile robot has been tailored for the purpose of the teaching aid. As has been already mentioned in subsection 2.4, the robot is equipped with three types of sensors: the reflective sensors; the magnetometer sensor; and the encoders. It is obvious that capabilities of the robot are determined by data provided by the sensors. Placement of a sensor, limitation of the hardware, type of output signal, or data acquisition; all these aspects define the measured data. For that reason, all these aspects are more than relevant for design of the control system of the robot; and consequently, they are detailed in following subsections.

3.1 Reflective Sensors

A reflective sensor used in the robot comprises of one infrared emitter and one phototransistor (a detector). The emitter side of the sensor is just the infrared emitter with an appropriate current-limiting resistor. The detector side is a resistor-capacitor circuit, where the resistance comes from the phototransistor. The resistance of the phototransistor is a measure of the incident infrared light. Namely, time of voltage decay t_d is measured after withdrawing of an external power supply from the resistor-capacitor circuit. Shorter time decay is an indication of greater reflection.

In the presented solution, QTRSensors library, which is provided by company Pololu Corporation, has been used for acquisition of the data. The library directly returns the time decay as the output. Range of the output signal is 0 µs to 2000 µs. Reading of the sensor data is performed with frequency 20 Hz.

The presented reflective sensors are appropriate for edge detection and line following; however, they are aimed to be used by the control system of the robot in a somewhat less conventional manner. Explicitly, the two front sensors are aimed to avoid a head-on collision with an obstacle. The side sensors are primary aimed for maintaining the desired direction of robot motion, i.e. they prevent the robot from running into an obstacle. In both cases, a distance l between a sensor and an obstacle is required to be estimated on the basis of the time decay.

In the maze, two types of obstacles can occur: partitions, and posts. As is obvious in Fig. 2, there are some blind zones in the observed area. These blind zones limit the use of the sensors. Due to the small size of the posts, the posts might be invisible for the robot. As is obvious in Fig. 3, no such limitations exist in the case of the partitions. Since the posts are always part of a wall, where most of the wall consists of partitions, the probability of a head-on collision with a post is very small. Thus, estimation of distances between the robot and posts is not carried out. Still, knowledge about distances between the robot and posts is relevant for maintaining the desired direction of motion.

Estimation of the distance l using the reflective sensors requires some calibration according to expected conditions of use. For that purpose, several experiments were carried out, and dependence of the time decay t_d on the distance l between a sensor and an obstacle was explored. All the experiments were performed under constant lighting condition (without any ambient light). The measurements were done for each sensor on five samples of each type of obstacles. The samples were chosen randomly. The obstacles were placed at the distances $l \in \{0, 1, 2, \dots, 40\}$ so that they were perpendicular to a beam emitted by a sensor. The distance l is given in cm.

On the basis of the measured data, a static characteristic for each sensor, and each type of obstacle, have been gained. For the partitions, the static characteristics are approximately linear for distances l in the range 2 cm to 32 cm for all four sensors. Slopes of the lines ensure sufficient resolutions of the sensors. The experiments have also shown that the time decay t_d is not sensitive on potential diversities in the coating of the partitions. Unfortunately, no such results have been acquired for the posts. No clearly visible dependence of t_d on l has been observed for l from 0 cm to 40 cm. The inapplicability of the signal for estimation of distances between the robot and posts is likely caused by high reflectivity of aluminum.

3.2 Magnetometer Sensor

The used magnetometer sensor allows sensing of magnetic fields in three axes. It is aimed for localization of the robot in the maze. In the teaching aid, a localization system introduced by Škrabánek and Vodička (2016) is employed. As Škrabánek and Vodička (2016) showed, the magnetometer sensor, in combination with magnetic landmarks, assures a sufficiently accurate and reliable localization of the robot.

The landmarks are used for labeling of cell borders. One landmark is placed in the middle of each border such that the longer side of the landmark is parallel with the border. Such placement ensures that the robot will not miss any landmark. In the maze, landmarks of type A have been installed according to specifications given by Škrabánek and Vodička (2016).

Since Škrabánek and Vodička (2016) have described technical parameters of this localization system in details, only the most relevant information has to be given in this paper. Thus, reading of the sensor data is ensured by the Adafruit_HMC5883_U library. For all three axes, the library returns magnetic fluxes B where the field range is 800 µT to −800 µT.

3.3 Encoders

In this robot, incremental optical encoders, based on reflective object sensors, are used. Namely, single-channel tachometer encoders have been built into the robot. The opaque/transparent patterns of the encoders consist of 38 equally sized divisions. It means that these encoders can sense a turn of the wheels of about $\frac{\pi}{19}$ rad. Since diameters of the wheels are about 6.1 cm, one pulse generated by an encoder corresponds approximately to 1 cm circumscribed by a particle on a wheel circumference. The reflective object sensors are connected with the microcontroller board via digital pins. Thus, no specialized library is needed for data acquisition in this case. Reading of the pulses is performed directly in the main control loop. A pulse is generated by a change of state on a pin.

4. ANALYSIS OF THE CONTROL PROBLEM

The role of the robot in the teaching aid is the execution of a path-plan scheduled by the high-level control system. The path-plan provided by a path-planning routine is expected to be a sequence of vertices the robot should pass through. However, the robot is not able to process path-plans provided in such form. The hardware equipment of the robot does not allow it to store a map of the maze in its memory. In other words, the robot does not dispose of any explicitly expressed map. For this reason, a conversion of path-plans to a more suitable form has to be carried out by the high-level control system.

In our solution, a conversion based on a notion of a safe and reliable movement of the robot through the maze has been used. From this perspective, keeping the robot at the greatest possible distances from walls of the maze seems to be the best approach. Such kind of movement is described by a trajectory, composed of straight-line segments, connecting centers of cells. An example of such trajectory is shown in Fig. 4 where the trajectory is depicted as a red dotted line.

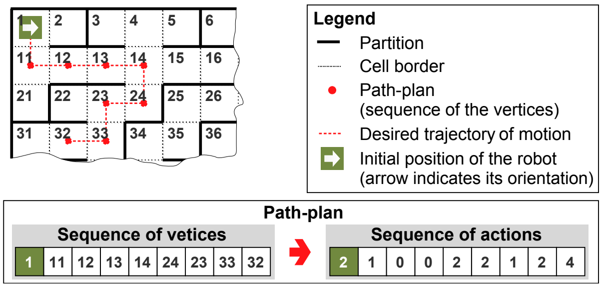

Fig. 4. Example of a path-plan and its transformation

Movement of the robot through a maze according to such trajectory means that the robot performs two kinds of motions related to the following performances: change of orientation, and shift of the robot from a current cell to an adjacent cell. The corresponding motions are a spot turn and a linear displacement of the robot. Just to clarify, the motions are always run separately. Although the linear displacement of the robot does not require any deeper analysis, the change of orientation is worth a deeper consideration. Clearly, the change of orientation is the key to an efficient conversion of the path-plan.

Let us consider the change of orientation from a process control point of view. Once the robot reaches the center of a cell, one of the following actions is performed: 'keep direction', 'turn left', 'turn right', 'turn back', and 'stop'. These actions can be expressed as an angle θ_R, where θ_R corresponds to an angle the robot should turn, from its original orientation, in order to reach a desired swing. On the condition that the robot receiving the instruction 'stop', it should not turn. Thus, it holds that $\theta_R \in \{0, -\frac{\pi}{2}, \frac{\pi}{2}, \pi\}$. For the sake of completeness, the orientation of the robot is referred to the face of the robot, where the face is a panel with reflective sensors (see Fig. 2). It is obvious that the just established actions can be used for the conversion of the path-plans into a more suitable form. For that purpose, codes stated in Table 1 are used. For a better clarity, also the angle θ_R is specified for each action in this table.

Table 1. Code table of actions

action	keep direction	turn left	turn right	turn back	stop
code	0	1	2	3	4
θ_R	0	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	π	0

In order to perform the transformation, orientation of the robot in the initial (actual) position has to be known. As was suggested by Škrabánek and Doležel (2014), this information can be obtained from the data provided by the camera system. For a better understanding, the process of the transformation is demonstrated in Fig. 4. The arrow in the green square symbolizes initial orientation of the robot. Red points are used to highlight the original path-plan (sequence of vertices). The original path-plan is also stated in the bottom part of the figure on the left. The path-plan after the conversion is shown on the right. It is obvious that the conversion transforms the sequence of vertices (cells) to a *sequence of actions*. The length of a sequence of actions is equal to the length of the corresponding sequence of vertices.

Implementation of a transformed path-plan by the robot then consists in sequential execution, of actions given by the path-plan. The robot executes just one action in each cell. After its execution, the robot moves straight to another cell, where this next cell is clearly determined by the robot's orientation after the execution of the action. In order to safely reach the center of the next cell, the correct localization of the robot in the maze has to be ensured.

As follows from this analysis, the control system of the robot has to guarantee proper implementations of path-plans. However, reading of sensor data, data evaluation, or generation of control actions has to be carried out by this system, too. The amount of activities to be carried out by the control system of the robot pushed us to divide the control system into two levels.

The higher level, which will be further called '*supervising level*', supervises implementations of path-plans. This is naturally associated with the localization of the robot. Thus, the localization is also carried out at this level. This level is detailed in section 5. The lower level of the control system ensures all the remaining functions, which have been mentioned in the previous text. Since these functions are closely related to the execution of the robot's movement, this level will be further called the *executive level*. It is described in section 6.

5. SUPERVISING LEVEL

As has been explained in section 4, a path-plan is provided to the robot as a sequence of actions to be executed in cells located on the path. The robot executes one action in each cell positively determined by the plan. Once the action is accomplished, the robot moves to a next cell. The next cell is one of the adjacent cells and it is explicitly given by the orientation of the robot after execution of the action. Transfer of the robot between two adjacent cells is required to be the linear displacement from the center of a current cell to the center of the next cell. It means that the trajectory is a line segment of length d connecting centers of the cells, where the length d is given by the size of cells, i.e. $d = 184$ mm.

As follows from the analysis in section 4, the robot is expected to perform two kinds of motions: the spot turn, and the linear displacement. From the perspective of process control, the robot should carry out one of two possible operations while executing a path-plan. However, such an approach does not cover the localization of the robot in the maze. As has been mentioned in subsection 3.2, localization of the robot is associated with the detection of cell borders. The localization of cell borders is performed within the linear displacement, hence this operation has been split up to two operations: exit from a current cell, and entrance to an adjacent cell. It means that three operations are considered at the supervising level: '*change orientation*' (execute an action), '*exit from a current cell*', '*entrance to an adjacent cell*'.

In order to execute a path-plan, these three operations are performed in a loop. In this loop, the operations are executed sequentially in a given order. Once the robot is initialized, the loop is opened with the operation 'change orientation'. The loop is run until all actions defined in a just performed path-plan are accomplished; or the instruction 'stop' is required to be implemented. As inherently follows, the loop terminates by the operation 'change orientation'. Once the loop is interrupted, the robot stops and waits for a reactivation by the high-level control system. The loop can be charted as is shown in left diagram of Fig. 5. The order of the operations is noted in the upper left corner of each pictogram. Since the sensors are essential for execution of the operations, their activities within each operation are also shown in this diagram; nonetheless, equally important is a regular update of the path-plan. For that reason, activity of the communication module is not omitted in this diagram.

As is well known, encoders are usually the backbone of a navigation system of a mobile robot. This robot will not be any exception. In the presented solution, the encoders embedded in the robot are necessary for execution of any of the operations. However, this is not valid for the remaining sensors. The reflective sensors are aimed to prevent the robot from collision within the liner displacement, and the magnetometer sensor is aimed for detection of cell borders. Thus, these sensors are not useful within the operation 'change orientation'. As is obvious in the left diagram in Fig. 5, the communication module is activated only within the operation 'entrance to an adjacent cell'. Communication of the robot with the high-level control system will be covered somewhat later.

The left diagram in Fig. 5 is primary aimed to show the control sequence employed at the supervising level. Nevertheless, it does not allow fully captured employment of the sensors within the operations related to the linear displacement. In order to get a closer idea about their usage, a diagram shown right in Fig. 5 has been constructed. In this diagram, activities of the sensors, and of the communication module, are shown at position d, where d is the position of the robot on the trajectory connecting two adjacent cells. The position of the robot is referred to by its geometric center and it is expressed as percentage of the distance covered by the robot from the beginning of the trajectory. Relations of the operations to the position of the robot are shown in the upper part of the diagram.

As follows from the facts stated in subsection 3.3, the encoders embedded in the robot have fairly low resolution which may result in a significant absolute position error. However, this drawback can be partially relaxed by resetting of the encoders in some appropriately chosen situations. Since the encoders have to be used both by change of robot's orientation and within the linear displacement, the absolute position error grows quite quickly. Considering these facts, the encoders are reset in two situations: before execution of the operation 'change orientation', and by crossing over a cell border. Reset of the encoders before a rotation of the robot is aimed to provide better performance while changing the robot's orientation. Their reset by crossing over borders serves primarily for localization of the robot, and consequently for 'timing' of the operations as is shown in Fig. 5. As follows from this diagram, the reflective sensors and the magnetometer sensor are active throughout the linear displacement.

Communication with the high-level control system is ensured by the Bluetooth module. Through this communication channel, a path-plan, or its part, can be sent to the robot. In our solution, the robot initializes the communication after entering to a next cell whereby the robot asks about potential updates of a current path-plan. Noteworthy is that the Bluetooth module is not activated until more than 50 % of the robot body is physically in the adjacent cell. This is important for a reliable detection of the robot position by the camera system; and consequently, for the correct updating of the path-plan. Activities of the communication module are shown in both diagrams in Fig. 5.

6. EXECUTIVE LEVEL

As follows from the name, this level is the executive part of the robot's control system, which ensures physical executions of path-plans. The physical execution of a path-plan is performed by the drivers. Control of the drivers is detailed in subsection 6.1. Accomplishment of a path-plan is managed by the supervising level. At this level, three different operations are considered. Given by the nature of the problem, different control strategies have to be used for different operations in order to safely execute the expected motions. This issue is discussed in subsection 6.2. As follows from the above stated, a proper functioning of the supervising level is guaranteed only as long as the right position of the robot in the maze is known. Although the localization of the robot is carried out at the supervising level, data necessary for the localization is provided by the executive level. This topic is considered in subsection 6.3.

6.1 Control of Drivers

The two drivers are the only actuators of the robot. Speeds of their rotations are controlled using a pulse-width modulation (PWM) technique. It means that the rotation speed is influenced by a PWM frequency and by a duty cycle D. In the presented solution, the PWMs share one timer, which is run in a fast PWM mode with a default setting of the PWM frequency, i.e. the frequency of the PWM signal is about 490 Hz. Since the frequency is not allowed to be changed by the control system, rotation speeds can be influenced only by the duty cycles D, which can be set via pins 9 and 10, i.e. $D \in [0, 255]$. Directions of rotations are controlled separately via two-valued pins 12 and 13. It means that a direction of a rotation is determined by a two-valued variable r_d, where $r_d = 1$ represents the clockwise, and $r_d = 0$ the counterclockwise direction of the rotation. Thus, two control variables, D and r_d, are available for each driver.

A desired output voltage of the PWM signal in 'the switch on state' is about 5 V; however, the actual value is influenced by a battery charge status. This naturally influences the speed of rotation. Luckily, both drivers are approximately equal affected by the state of the battery. Since the speed of robot movement is not crucial, the rotation speeds do not have to be maintained at predetermined levels. For that reason, the influence of

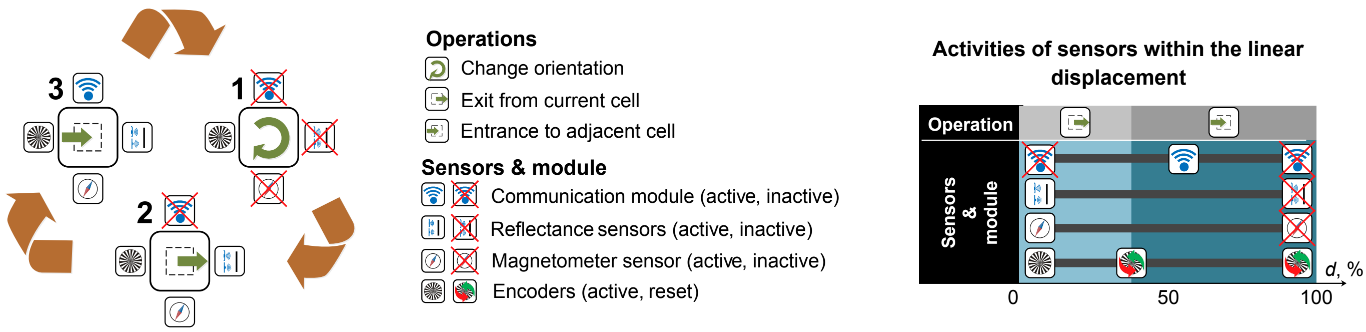

Fig. 5. On the left: control sequence executed on the supervising level; on the right: usage of sensors according to position of the robot within the linear displacement (relative to the geometric center of the robot)

battery charge status on rotation speeds is neglected. In other words, it is assumed that the rotation speed of a driver is, for a particular setting of the duty cycle D, constant in time.

6.2 Control Strategies

As has been stated in section 4, path-plans are provided to the robot by the high-level control system as sequences of actions to be carried out in cells. In order to execute a plan effectively and safely, a trajectory composed of straight-line segments, connecting centers of cells, is required to be followed by the robot. The correct implementation of path-plans is ensured by the supervising level. From the perspective of the supervising level, the execution of a path-plan consists of the proper implementation of three operations. A physical implementation of these operations is carried out by the executive level. It is evident from the nature of the operations that different control strategies have to be used for different operations. Although three operations are considered at the supervising level, essentially only two types of activities are carried out by the robot: the change of orientation, and the linear displacement. Thus, only two control strategies are needed by the implementation of the operations. These strategies are detailed in the following text.

Strategy 1: Change of Orientation Basically, the robot turns on the spot about a predetermined angle θ_R within the operation 'change orientation'. Since the robot should turn on the spot, the wheels should turn the same speed; however, in opposite directions. In order to reach a desired swing, the wheels have to be turned about a predetermined angle φ. The only sources of information about rotations of the wheels are the encoders. As follow from subsection 3.3, the outputs of the encoders can be considered as generators of pulses. Counting number of pulses, generated by an encoder, gives information about a wheel rotation. Considering this, a desired value of φ is expressed as a number of pulses n_e to be generated when implementing an action. Setting of the drivers, as well as the desired values of pulses n_e, is stated for all actions in Table 2.

Strategy 2: Linear Displacement A path, the robot should follow, consists of segment lines. It means that the robot should perform linear displacements between adjacent cells according to a path-plan. Under ideal conditions, rotations of the wheels through the same angle φ, the same speed, and in the same direction should be sufficient for

Table 2. Setting of the drivers and desired numbers of pulses for the actions performed within the operation 'change orientation'

action	0		1		2		3		4	
driver	L	R	L	R	L	R	L	R	L	R
D	0	0	90	90	90	90	90	90	0	0
r_d	1	1	0	1	1	0	0	1	1	1
n_e	0	0	6	6	6	6	13	13	0	0

the transfer of the robot from a current cell to an adjacent cell.

At first, let us consider φ, the angle about which the wheels should turn in order to execute the linear displacement. As has been mentioned in section 5, the linear displacement has been divided into two operations. Since the end of the operation 'exit from current cell' is clearly determined by detection of a cell border, a desired value of φ does not have to be defined for this operation. Thus, a desired value of φ, or more specifically a desired value of pulses n_e, has to be determined only for the operation 'entrance to an adjacent cell'. Considering the size of cells, and the construction and the dimensions of the robot, the robot should cover about 15 cm to reach the center of a next cell. Thus, 15 pulses might be generated within this operation, i.e. $n_e = 15$ for both wheels.

In order to prevent the robot from any collision, the robot has to move only forward; therefore, directions of the driver rotations are clockwise for both drivers, i.e. $r_d = 1$. As has been stated in subsection 3.1, assessment of a collision risk rate is based on data provided by the front reflective sensors. Once reflectance measured on one of these sensors is under 60 µs, the robot is stopped completely and a human intervention is expected in order to resume working of the robot.

Under ideal conditions, rotation of the drivers at the same speed might be sufficient to move straight on. Unfortunately, the real conditions are far from the ideal. Any inaccuracy of an encoder, any unevenness of the floor, or any asymmetry of a wheel may cause deviation of the robot from a desired path. Once the robot swings significantly away from the path, the direction of the robot movement has to be corrected.

The only way to detect a skew is to use data provided by the side reflective sensors. As has been shown in subsection 3.1, the sensors can be used for estimation of a

Table 3. Expected intervals of outputs of side reflective sensors according to the position of the robot within a cell and the configuration of side partitions

		R_L		R_C		R_R	
p_L	p_R	t_{d_L}	t_{d_R}	t_{d_L}	t_{d_R}	t_{d_L}	t_{d_R}
0	0	$[151, 2000]$	$[141, 2000]$	$[151, 2000]$	$[141, 2000]$	$[151, 2000]$	$[141, 2000]$
0	1	$[151, 2000]$	$[81, 140]$	$[151, 2000]$	$[40, 80]$	$[151, 2000]$	$[0, 39]$
1	0	$[0, 38]$	$[141, 2000]$	$[39, 80]$	$[141, 2000]$	$[81, 150]$	$[141, 2000]$
1	1	$[0, 38]$	$[81, 140]$	$[39, 80]$	$[40, 80]$	$[81, 150]$	$[0, 39]$

distance between a sensor and a partition; but a distance between a sensor and a post cannot be estimated due to the high reflectivity of the posts.

As follows from the previous text, the position of the robot within a cell can be estimated once a partition is presented on a side border of the cell, where the side borders are determined by a current orientation of the robot in the cell. In other words, side borders are the borders, which are sensed by the side reflective sensors of the robot. Nevertheless, sensitivity of these sensors allows only rough estimation of the robot's position within a cell. For that reason, only three positions of the robot, related to the sides of cells, can be positively identified: the robot is in the middle of the cell (R_C); the robot is close to the left (R_L) border; the robot is close to the right (R_R) border of the cell.

Considering all potential configurations of partitions on the side borders, four possible scenarios can occur: only the left partition is presented; only the right partition is presented; both side partitions are presented; and no side partition is presented. Let us express presence of a partition on a cell border as a binary variable p, where $p = 1$ denotes presence and $p = 0$ implies absence of a partition. For the left partition, variable p_L will be used, and similarly, for the right partition, variable p_R will be used.

For all these scenarios, dependence of the outputs of the side reflective sensors on the position of the robot within a cell has been examined. For that purpose, position of the robot in a cell has been determined in the following way: the robot is considered to be in the middle of a cell when distances to both side borders of the cell are greater or equal to 2.5 cm.

On the basis of the experiment results, expected intervals of time decays t_d have been determined for each possible combination of 'robot position'-'partition configuration'. The obtained results are summarized in Table 3. In the first two columns of the table, configurations of partitions are given. In the remaining columns, intervals of time decays are stated. The position of the robot within a cell is indicated in the first row of the table.

Since only three positions of the robot within a cell are distinguished, only three control actions come into consideration. In the case that the robot is in the middle of a cell, the robot should keep direction (KD) of the movement. In the case that the robot is near to the left border, the robot should turn right (TR). In the case that the robot is near to the right border, the robot should turn left (TL). However, the position of the robot is not always known. In a such situation, the robot cannot make any relevant decision. However, it can be assumed that the robot moves in the correct direction. Thus, it seems to be the best practice, in such situations, to keep the current direction of the movement.

Summarizing these rules into a lookup table, a table based controller has been created. The lookup table is shown in Table 4. As is obvious, the standard arrangement of lookup tables has been modified in order to provide better readability. The colors used within the cells of the table express the position of the robot in a cell of the grid formed over the maze. The light blue color denotes that the robot is close to the left border; however, the orange means that that the robot is close to the right one. The green color symbolizes that the robot is in the middle of a cell. The salmon pink is then used for a situation when the position of the robot is indeterminable.

Table 4. Lookup table for linear displacement

$t_{d_L} \backslash t_{d_R}$	$[0, 39]$	$[40, 80]$	$[81, 140]$	$[141, 2000]$
$[0, 38]$	KD	KD	TR	TR
$[39, 80]$	KD	KD	KD	KD
$[81, 150]$	TL	KD	KD	TL
$[151, 2000]$	TL	KD	TR	KD

The control actions stated in Table 4, are realized by changing of duty cycles of the PWMs. Appropriate settings of D are summarized in Table 5 where abbreviations of the control actions can be found in the first row. In the second row, duty cycles for the left driver D_L are given. Duty cycles for the right driver D_R are the listed in the last row.

Table 5. Setting of PWMs for the control actions

action	KD	TL	TR
D_L	90	50	120
D_R	90	120	50

6.3 Detection of Cell Borders

The detection of cell borders is key for the localization of the robot in the maze. As has been mentioned in subsection 3.2, the detection primarily relies on the magnetometer sensor and the landmarks. Škrabánek and Vodička (2016) recommend to use simultaneously magnetic fluxes measured in the sense of the y and z axes in order to ensure the most reliable detection of cell borders. However, it has been found during our experiments with the robot in the maze that only data measured in the sense of the y axis are valuable. In the sense of the z axis, some unexpected signals have been randomly detected in a large distance from the landmarks.

Luckily, the loss caused by the drop out of the z axis can be fully compensated by data provided by the side reflective sensors. Namely, the high reflectivity of the posts can be used for this purpose. Just to clarify, one post is placed in each corner of a cell. It means two posts are on each border of each cell. When the robot passes between two posts, time decays measured by the side reflective sensors might be close to zero, i.e. $t_{D_L} \in [0, 38]$ and $t_{D_R} \in [0, 39]$. A simultaneous decline of the magnetic flux under $-300\,\mu T$, in the sense of the y axis, positively determines that the front of the robot is just crossing a border of a cell.

7. CONCLUSION

In this paper, the control system designed for the semi-autonomous mobile robot has been presented. The mobile robot has been developed as a part of the teaching aid which has been also reflected in the construction of the robot. The robot is based on the microcontroller board Arduino UNO. The robot is equipped by four reflective sensors, two encoders, and one magnetometer sensor. These sensors are sufficient for the successful navigation and localization of the robot in its workspace. Naturally, the workspace has been designed with respect to the robot's equipment. In such a way, an inexpensive teaching aid, appropriate for practicing the path-planning theory, has been created.

The role of the robot in teaching is the execution of path-plans scheduled by the high-level control system. Moreover, the path-plans should be implemented by the robot autonomously. It means that the control system of the robot has to carry out data acquisitions, data processing, control actions generation, as well as a path-plan implementation, and the localization of the robot. These activities can be dived into two groups which have been reflected in the structure of the presented control system.

The control system of the robot has been divided into two levels. Roughly speaking, the higher level is aimed to supervise the correct implementation of a path-plan, while the lower level ensures the physical execution of the path-plan. The higher level has been simply conceived as a sequential machine, periodically performing three operations. According to an operation being performed, an appropriate control strategy is applied on the lower level. In order to ensure successful execution of the operations, data provided by the sensors has to be treated in an appropriate way. The multi-sensor integration has been discussed, too.

The key for the successful implementation of the presented control system is the reliable localization of the robot in the workspace, which is represented by a maze. In the presented solution, the localization system introduced by Škrabánek and Vodička (2016) is employed. However, it could not be used fully in its original form. In the maze, data measured in the sense of the z axis is burdened by an error. Specifically, the magnetic field might be detected at a large distance from the landmarks in this axis. The probable grounds are metal components used in construction of the maze. In order to cover the loss of this signal, data provided by the side reflective sensors are used. Namely, the high reflectivity of the posts is used for this purpose. The modified localization system has proven to be reliable and accurate.

From a programming standpoint, the construction of the robot has proven to be very convenient. Support of the Arduino platform by the user community is at an excellent level. The freely available libraries have greatly facilitated the development of the control system of the robot. The presented control system has been implemented in the robot, and it has been verified in several mazes of different layouts. Under standardized lighting conditions, the robot was able to execute given path-plans without any collision. From these experiments, Vodička (2016) captured several videos. As is obvious, the presented solution is functional and it is sufficient for the educational purpose.

A certain disadvantage is the need to operate the robot under standardized lighting conditions. As follows from our experiences, sunlight may lead to a failure of the control system. This is caused by the infrared component of the sunlight, which affects the reflective sensors. This imperfection might be overcome using frequency sweep by obstacle detection. Its implementation will be considered in the next version of the control system.

REFERENCES

Alves, S.F.R., Ferasoli Filho, H., Pegoraro, R., Caldeira, M.A.C., Rosário, J.M., and Yonezawa, W.M. (2011). *Research and Education in Robotics - EUROBOT 2011*, chapter Educational Environment for Robotic Applications in Engineering, 17–28. Springer Berlin Heidelberg.

Cuéllar, M.P. and Pegalajar, M.C. (2014). Design and implementation of intelligent systems with LEGO mindstorms for undergraduate computer engineers. *Computer Applications in Engineering Education*, 22(1), 153–166.

Harlan, R.M., Levine, D.B., and McClarigan, S. (2001). The khepera robot and the krobot class: A platform for introducing robotics in the undergraduate curriculum. *SIGCSE Bull.*, 33(1), 105–109.

Mariška, M. and Doležel, P. (2014). Multi agent environment for modelling and testing of cooperative behaviour of agents. *Advances in Intelligent Systems and Computing*, 289, 301–306.

Pakdaman, M. and Sanaatiyan, M.M. (2009). Design and implementation of line follower robot. In *Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on*, volume 2, 585–590.

Rubenstein, M., Ahler, C., Hoff, N., Cabrera, A., and Nagpa, R. (2014). Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Robotics and Autonomous Systems*, 62(7), 966–975.

Vodička, P. (2016). Operation of the robot in the maze (videos). http://www.upce.cz/fei/krp/zaverecne-prace/bp-obhaj4/bp-vodicka.html.

Škrabánek, P. and Vodička, P. (2016). Magnetic strips as landmarks for mobile robot navigation. In *Information Technology Convergence, 2007. ISITC 2007. International Symposium on*. Unpublished.

Škrabánek, P. (2015). Labyrinth arrangement analysis based on image processing. In *Mendel 2015*, volume 378 of *Advances in Intelligent Systems and Computing*, 305–316. Springer International Publishing.

Škrabánek, P. and Doležel, P. (2014). Attractive robot's design suitable for image processing. In *Proceedings of the Mendel 2014: 20th International Conference on Soft Computing*, 217–222. University of Technology Brno.

Škrabánek, P., Mariška, M., and Doležel, P. (2015). The time optimal path-planning of mobile robots motion respecting the time cost of rotation. In *Proceedings of the 20th International Conference on Process Control*, 232–237. STU Bratislava.