Design of Machines and Structures, Vol. 13, No. 2 (2023), pp. 45–52.
https://doi.org/10.32972/dms.2023.016

DEVELOPMENT OF A TEST BENCH TO SOLVE A SIMPLE LABYRINTH

PÁLMA KAPITÁNY[1] – MÁTÉ SZABÓ[2]

University of Miskolc, Robert Bosch Department of Mechatronics
3515, Miskolc-Egyetemváros
[1]palma.kapitany@uni-miskolc.hu, [2]szabo.mate1@student.uni-miskolc.hu
[1]https://orcid.org/0000-0001-6826-2371

Abstract: This paper deals with building an automatic test bench for detecting and solving a simple maze. There are many suitable sensors for detecting walls and ball, as find quick and stabile algorithm for deliver the ball to the target. Two servo motor are used to tilt the table of the maze. The first part of the paper shows the experiences of using OV7076 camera with Arduino Nano which was used for the first time to solve the problem. The final version of the maze solver contains a USB camera, an Arduino UNO platform, OpenCV and PyCharm software. Furthermore, paper reveals possibilities of decision-making-process and the direction of following research work.

Keywords: *maze, OpenCV, PyCharm, ball labyrinth*

1. INTRODUCTION

Nowadays one of the leading sectors is the development of self-driving vehicle (Almadhoun & Zhang, 2021), (Kosuru & Venkitaraman, 2023), the use of cooperative robots (Yan, Zeng, He, & Hong, 2023), (Yu, Tong, Xu, Dong, & Yang, 2021) and a wide range of autonomous decision-making devices. These kinds of tasks can serve convenience functions, help people, or operate autonomous systems, thus freeing up human resources and freeing people's time for creative work. In the case of such systems, sensor signals are used in a targeted manner, decisions are made in the light of the defined goal, and the necessary actuators are controlled. Many self-driving vehicles and autonomous system must solve a maze-situation, discover, and follow possible path (Mohammed, Al-Dabagh, & Rashid, 2023), (Suryani, Agustriana, Rakhmatsyah, & Pahlevi, 2023).

This article also presents an independent system, the stated goal of which is to be suitable for solving a simple ball maze, based on image processing, and to come to rest when the ball reaches the target. Section 2 deals with structure of ball maze solving system and shows the way of moving table of labyrinth, the 3D model of the plan was made in the Autodesk Inventor software.
Section 3 introduces the first solution that used an OV7670 camera and Arduino Nano development platform with ATmega328 microcontroller. Section 4 discusses elements and control of the final system. The solution includes USB camera used in OpenCV, cooperate with PyCharm to control two servo motors via Arduino Uno platform. Several algorithms are known, which can be used to get out of a labyrinth. In the case of the simple ball maze, reaching a target position is desirable, therefore the ball must be roll along the path leading to it.

2. STUCTURE OF BALL LABYRINTH SOLVING SYSTEM

The maze is placed on a double table, which contains two sheets. The top sheet holds the ball and the walls of the maze, at the ends a hinge is opposite to a servo motor, these two elements provide the attach to the bottom sheet. Figure 1 shows that if Motor-A is turn counter-clockwise, the left side of top sheet rises therefore the ball rolls direction from the left to the right. The speed of the ball depends on the size of the slope of the sheet. Motor-A can turn clockwise, when the arm of servo let the sheet under the horizontal, ball rolls direction from the right to the left. The bottom sheet connects to the ground via two hinges (Hinge-B) which are standing opposite to Motor-B. When both motors are used at the same, there is possible to roll the ball in any direction of the plane, also it could be holding the ball in one position. Table 1. presents these cases.

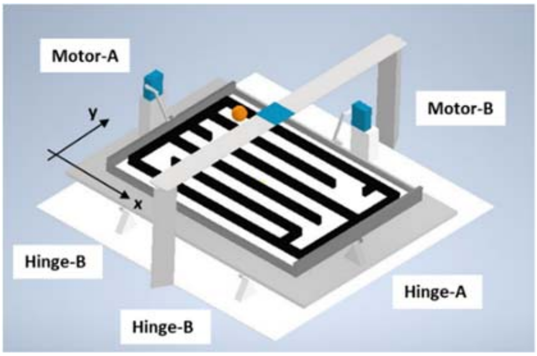

Figure 1. *Structure of ball labyrinth solving system (Szabó, 2023)*

In order to detect the ball and make decision to solve the maze problem, a camera is mounted onto the top of the system. The entire labyrinth can be checked via the camera unit. The planned structure included an electrical wiring for the microcontroller and the necessary electric power sources.

Table 1

Rolling direction of the ball depending on arms of servo motors

Motors	**B: Up**	**B: Horizontal**	**B: Down**
A: Up	x+ and y-	x+	x+ and y+
A: Horizontal	y-	ball in rest	y+
A: Down	x- and y-	x-	x- and y+

3. EXPERIENCE OF USING OV7670 CAMERA WITH ARDUINO NANO

In the first case, an OV7670 camera was used with Arduino Nano platform which includes ATmega328 microcontroller. The process was the following: wiring 18 pins of the camera to Arduino Nano pins, connection of pull-up and pull-down resistors, the use of voltage divider, common ground potential, data transfer cables and restart point. Thereafter the ATmega328 microcontroller is programmed in the Arduino IDE software on a personal computer. The program contains the following elements: setting of type of camera, numbers of scannable rows and columns, value of frames per second; next to checking a serial communication between the personal computer and Arduino Nano platform.

Furthermore, two servo motors were also connected to Arduino Nano platform by digital output pins in PWM mode, common ground GND potential pins, next to motors were wired to the necessary voltage supply.

In this phase of the research, it had dealt with building a faster connection between the microcontroller and the camera, as well as developing the software system during image processing. In addition, several methods and layouts were examined to create the wall of the labyrinth, as well as the sensitivity of the camera, which colours should be chosen as colour of the base plan, and which colour are contrasting for the walls and the ball.

OV7670 camera had 25° chief ray angle, therefore camera was placed the height of 700 mm from the sheet of labyrinth. Figure 2. part a) shows a photo about the real colours and lighting conditions, next to Figure 2. part b) illustrates the picture what was produced by scanning of OV7670 camera, sent by ATmega328 via Arduino Nano platform to the personal computer.

a) Photo by reality b) Scanned picture via OV7670

Figure 2. *Experience of using OV7076 camera with Arduino Nano (Szabó, 2023)*

Figure provides that this scanned picture is not suitable for servicing the input data for maze-solving system. Direction of developing of maze-solving system was changed after this first planned and pilot construction. Main goal was in this case to build a control based on the microcontroller separately from a computer. Meaning of the second phase of the research was to make a control based on the personal computer.

4. DETAILS OF THE FINAL SOLUTION

As pointed out in the previous section, the control is planned to move from the microcontroller to a personal computer. In this way significantly more possibilities were revealed from the software support side. During the development, the OpenCV software came to the fore, first used in programming language C, thereafter the final program is written in Python language. Later, the use of PyCharm software became justified for the real-time control of the microcontroller which set position of the servo motors. In addition, changes were also made on the hardware side: instead of the Arduino Nano platform an Arduino Uno device was built into the system.
Figure 3 shows the placement of the camera which monitoring the labyrinth, the personal computer that runs the necessary software support, furthermore it contains the Arduino Uno, and the servo motors in the structure of labyrinth solver. The right side of Figure 3 shows the test bench, which was built from hard paper; OV7670 is

replaced with a general USB camera, and it is placed on top of structure. This construction was semi-closed box because of testing separately from the lights of environment.

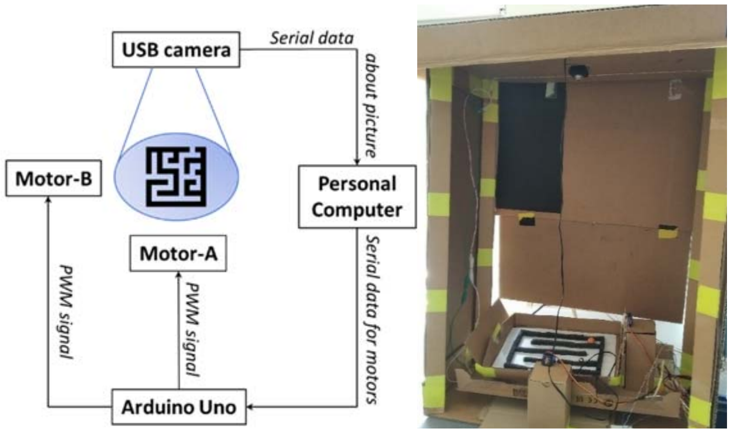

Figure 3. *Control of simple maze-solving system based on personal computer*

In this prototype lies a simple S-shaped labyrinth with a plastic ball and walls. Simple S-shaped track means that to get the ball to the target, it only requires a one-way tilt and a right-to-left roll. This control can be achieved by use two servo motors: one of them lifts the end of bottom sheet to tilt the table, the other lifts and let down the end of top sheet to roll the ball. White colour is used for the background of maze, dark grey (near to black) for the walls and orange for the ball.
The solver first needed to filter the image according to the colours mentioned above. After detecting the wall of the labyrinth, it surrounded its largest extent with a virtual rectangular contour and determined the coordinates of this rectangle. Using a similar procedure, the image processing system filtered the orange colour and identified the ball and can also characterize it with the coordinates of its centre.
After the separation by colour and the determination of the extents and the description with the coordinates, the decision-making and the control of the servo motors followed. The first mechanism for solving the simple S-shaped maze tilted the lower plate towards the target position in the upper left corner, while the second motor tilted it to the right if the ball was located on the left side, and vice versa, using X coordinate. Naturally where the wall was ended and a free path was opened to move in the direction of the goal, the ball continued on the s-shaped solving track all the way to the goal. Where the Y coordinate condition stopped the process and the ball remained in the top left target position.

Figure 4 shows the two tuned filters, with the help of which it identified the colour of the ball or the wall in the original image on the left, and cropped the image accordingly, and then the end shows the largest unit bounded by the filter, presented with a blue contour, and characterized by the centre coordinates.

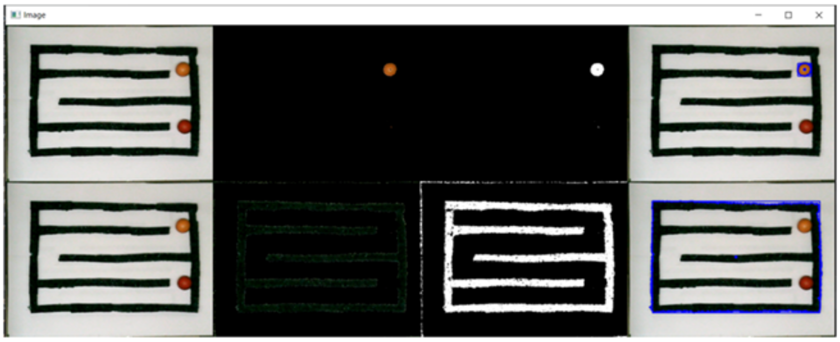

Figure 4. *Two implemented filter and detection of ball and wall (Szabó, 2023)*

Figure 4 and Figure 2 can be compared, there is a significant difference in the quality of the scanned image. It can also be cleared that the system was able to distinguish between shades of orange and red. A detailed description of the tuning of the filters can be found in the literature (Szabó, 2023), as well as how OpenCV and PyCharm were programmed to receive the camera image, process it and control the Arduino Uno development platform. The operation of the algorithms, the instruction content of the called command lines and the tuning of the parameters can also be read in that literature. The decision-making of the solver is currently based on the X and Y comparison of the centres of the largest contours identified with two different colours.

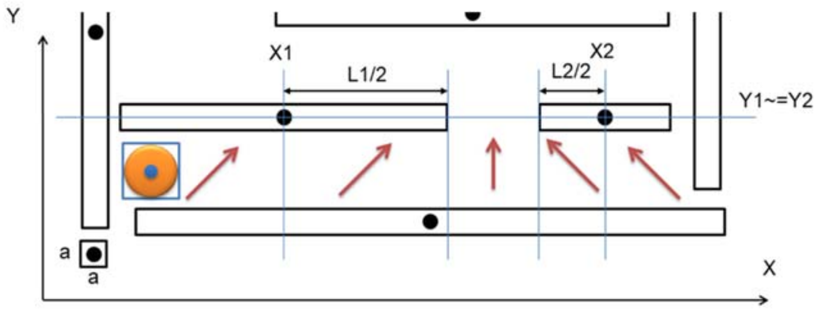

Figure 5. *The decision-making of the solver is currently further sub-units (Szabó, 2023)*

During the later development, it will be accessible by dividing the wall into further sub-units, and by making the gates in the walls a local goal: that the system is not only able to solve S-shaped labyrinths, but also general and medium difficulty mazes. Figure 5. shows a solution to take a decision process in direction X, it can be used in direction Y. This method requires an etalon, which is presented in Figure 5 by small letter a.

5. CONCLUSION

This article dealt with the design and construction of a maze-solving system capable of moving the ball from a single S-shaped maze to a predetermined target position. During the chapters, it reported on the results of an earlier research, which included a self-made image processing unit, which did not achieve the desired accuracy and repeatability. Paper shows on the work that moved the software requirement to other software that can be run on a personal computer and is able to communicate with motor-control development platforms.

Furthermore, it designates the direction of further development, which with the change brought about in the field of image processing: dividing the wall into further sub-units the specific shapes of the maze and leads the ball out of more than just a simple S-shaped maze.

REFERENCES

Almadhoun, R., & Zhang, W. (2021). Autonomous vehicle technology: A review of advancements, challenges, and future directions. *IEEE Transactions on Intelligent Transportation Systems, 22*(4), 2133-2149. https://doi.org/10.1109/TITS.2020.3005637

Kosuru, V., & Venkitaraman, A. (2023). Advancements and challenges in achieving fully autonomous self-driving vehicles. *World Journal of advanced Research and Reviews , 18*(1), 161-167. https://doi.org/10.30574/wjarr.2023.18.1.0568

Mohammed, I., Al-Dabagh, M., & Rashid, S. (2023). Path discovering in maze area using mobile robot. *TELKOMNIKA Telecommunication, Computing, Electronics and Control., 20*(2). https://doi.org/10.12928/telkomnika.v20i2.19408

Suryani, V., Agustriana, K., Rakhmatsyah, A., & Pahlevi, R. (2023). Room cleaning robot movement using A* algorithm and imperfect maze. *JURNAL INFOTEL, 15*(1), 75-81. http://doi.org/10.20895/infotel.v15i1.901

Szabó, M. (2023). Automata golyólabirintus megoldó tervezése. *BSc. szakdolgozat*. Miskolci Egyetem.

Yan, X., Zeng, Z., He, K., & Hong, H. (2023). Multi-robot cooperative autonomous exploration via task allocation in terrestrial environments 17:1179033. *Frontiers Neurorobot, 17*. https://doi.org/10.3389/fnbot.2023.1179033

Yu, J., Tong, J., Xu, Y., Dong, H., & Yang, T. (2021). SMMR-explore: submap-based multi-robot exploration system with multi-robot multi-target potential field exploration method. 8779-8758. https://doi.org/10.1109/ICRA48506.2021.9561328