

Case Study on Creating Flowcharts for Mobile Robot Line Follower Using Robomind Simulator and Non Simulator Fischertechnik

Aulia Sabril¹, Sutarsi Suhaeb²
Universitas Negeri Makassar, Indonesia

aulia.sabril@unm.ac.id¹

Kata Kunci:

Robotika, Pengikut Garis, Diagram Alur, Robomind, Fischertechnik, Simulasi, Non-Simulasi.

Abstrak

Pemrograman robot umumnya dimulai dengan membuat diagram alur untuk menentukan fungsi teknis robot sebelum menerapkannya menggunakan bahasa pemrograman seperti C, Ruby, atau lainnya. Namun, pendekatan ini memiliki kelemahan karena membutuhkan waktu, biaya, dan upaya yang signifikan, sehingga kurang efisien dan efektif. Salah satu solusi untuk mengatasi masalah ini adalah dengan langsung menggunakan diagram alur dalam proses pemrograman. Mainan edukatif berdasarkan Brick, seperti Fischertechnik, telah mengadopsi konsep ini, memungkinkan program menjadi lebih mudah dibaca, sistematis, mudah dimodifikasi, dan memfasilitasi identifikasi kesalahan. Penelitian ini bertujuan untuk membandingkan diagram alur yang dibuat secara manual di simulator Robomind dengan yang diimplementasikan menggunakan perangkat Fischertechnik non-simulator untuk fungsi yang sama, yaitu sebagai mobile robot line follower. Hasil penelitian menunjukkan perbedaan bentuk dan implementasi diagram alur antara kedua pendekatan ini. Untuk penelitian di masa depan, diusulkan untuk membandingkan lebih lanjut diagram alur yang dibuat di simulator Robomind dan langsung diunggah ke perangkat Fischertechnik non-simulator untuk tugas yang sama.

Key Word:

Robotics, Line Follower, Flowchart, Robomind, Fischertechnik, Simulation, Non-Simulation.

Abstract

Robot programming generally begins with creating a flowchart to define the robot's technical functions before implementing it using programming languages such as C, Ruby, or others. However, this approach has weaknesses as it requires significant time, cost, and effort, making it less efficient and effective. One solution to overcome this issue is by directly using flowcharts in the programming process. Educational toys based on Brick, such as Fischertechnik, have adopted this concept, allowing programs to be more readable, systematic, easily modified, and facilitating error identification. This study aims to compare manually created flowcharts in the Robomind simulator with those implemented using non-simulator Fischertechnik devices for the same function, namely as a mobile robot line follower. The study results indicate differences in the form and implementation of flowcharts between these two approaches. For future research, it is proposed to further compare flowcharts created in the Robomind simulator and directly uploaded to non-simulator Fischertechnik devices for the same tasks.

Copyright © 2025 Aulia Sabril, Sutarsi Suhaeb

This work is licensed under an **Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)**

INTRODUCTION

Line Follower Robot is a basic robotics project commonly used in robotics education to introduce concepts of robotic control and sensors. This robot utilizes sensors to detect lines and navigate along predetermined paths. Simulation approaches are becoming increasingly popular as they allow users to design and test robotic algorithms before implementing them on physical hardware. In this study, two different platforms are used: Robomind as a simulator and Fischertechnik as a non-simulator platform. This research aims to analyze the advantages and disadvantages of both platforms in developing a line follower robot and document the flowchart creation process for the algorithms used to control the robot on both approaches.

Materials and Methods Material Simulator

Robomind Software as a virtual simulation for testing line follower robot flowchart algorithms and logic.



Picture 1. Software Robomind

Non Simulator

- 1) The Fischertechnik Robotics kit is a device platform that is often used to physically learn. The main component are motors, camera, and controller.



Picture 2. Kit Robotics Fischertechnik

- 2) Software Robo Pro for programming Fischertechnik robotics kit.



Picture 3. Software Robo Pro

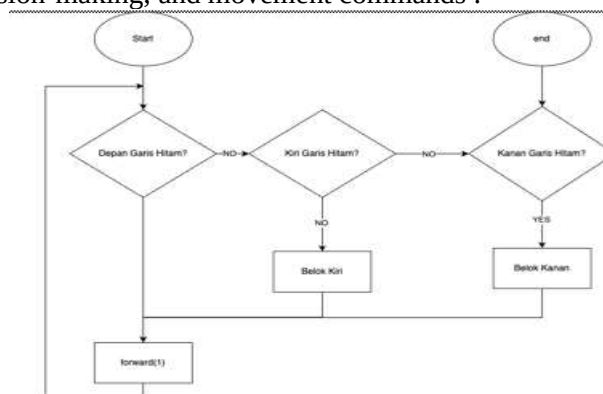
Komputer or Laptop

Used to create and upload programs.

METHOD

Making a Mobile Robot Line Follower Flowchart

The initial stage involved designing a flowchart for the mobile robot line follower, which generally includes line detection, decision-making, and movement commands .



Picture 4. Mobile Robot Line Follower Flowchart

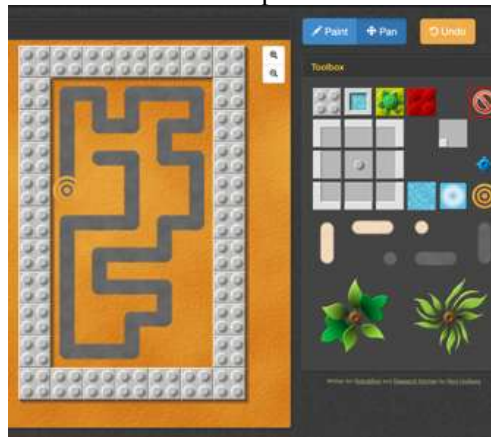
Simulation Process

Robomind is a simulation tool that allows users to program virtual robots using a simple programming language. The Implementation of the line follower in the Robomind simulator involves the use of virtual sensors to detect lines in the simulation arena.

Steps in Robomind:

Arena Configuration:

Configure the simulation arena with a black line path.



Picture 5. Arena Simulator Mobile Robot

Programming:

Determines how the sensor reads black line data.

Applying Flowchart Logic:

Applying flowchart logic to programming.

Simulation:

Run simulations to test robot behavior.



Picture 6. Simulation Process

Non-Simulator Process

The line follower robot built with Fischertechnik uses a camera sensor to detect lines and actuators (motors) to drive the robot.

Steps in Fischertechnik:

Robot Assembly:

Assemble the robot using Fischertechnik components, including motors and cameras.

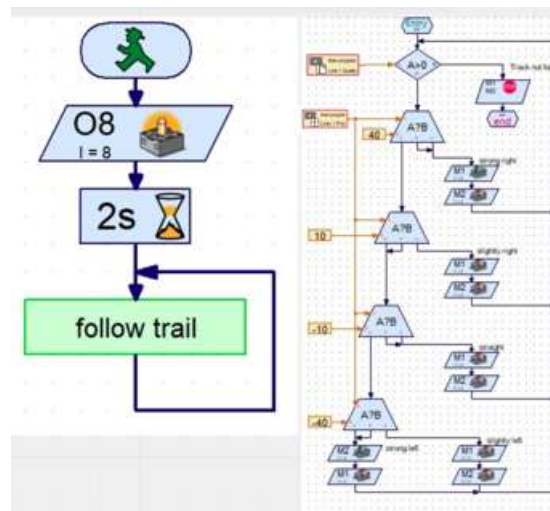


Picture 7. Mobile Robot Line Follower Fischertechnik

Sensor Calibration:

Configure the optical sensor, where the camera is used to detect the color of the line.

Programming:

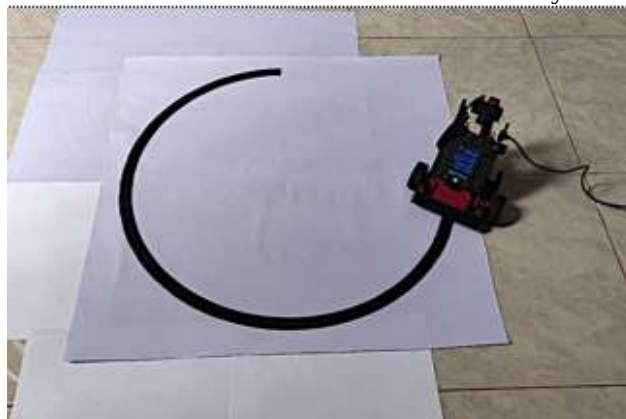


Picture 8. Flowchart Programming

Implement flowcharts using block-based programming or code with the Robo Pro program.

Testing:

The Robot was tested on a real track with black lines on a white vinyl cloth surface.



Picture 9. Testing Mobile Robot in arena

RESULT AND DISCUSSION

Simulator Robomind

On the Robomind simulation platform, the robot was able to follow lines quite well as all environmental conditions could be perfectly controlled. The simulator's sensors read the path with high accuracy, and the simulation results demonstrated that the robot could follow the line stably without physical disturbances such as friction or surface imperfections.

Non-Simulator Fischertechnik

On the non-simulation platform Fischertechnik, the results showed several more realistic challenges, such as the need for highly precise sensor calibration and mechanical challenges related to the robot's movement on real surfaces. Real-world testing revealed hardware limitations, such as motor instability and noise from the camera as a sensor, which needed to be addressed through additional tuning of light intensity settings and the size and width of the lines detected by the camera.

Discussion

Advantages of Simulation (Robomind). Robomind enables quick and flexible algorithm testing without requiring physical components, making it more time- and cost-efficient. The perfectly controlled environment provides ideal results without physical obstacles.

Advantages of Non-Simulation (Fischertechnik). Fischertechnik offers a more realistic experience and prepares users to face real-world challenges in robotics. Issues such as sensor calibration, motor control, and mechanical disruptions provide essential lessons that cannot be encountered in simulations.

CONCLUSIONS

The use of simulators and non-simulators in the development of line follower robots has its own advantages and disadvantages. Robomind offers flexibility and efficiency in the initial development stages. However, the use of Fischertechnik provides a more realistic experience, which is crucial for addressing mechanical and physical challenges often encountered in real-world conditions. The process of creating flowcharts for both platforms is fundamentally similar. However, their implementation exhibits significant differences, particularly regarding hardware adjustments.

REFERENCES

- Bekey, G. A. (2005). *Autonomous robots: From biological inspiration to implementation and control*. MIT Press.
- Bhattacharya, S. (2011). Sensor technology in robotics. *Sensors and Actuators Journal*.
- Bolton, W. (2015). *Mechatronics: Electronic control systems in mechanical and electrical engineering*. Pearson.
- Bradley, D., Seward, D., Dawson, D., & Burge, S. (2000). *Mechatronics and the design of intelligent machines and systems*. CRC Press.
- Bruno, J. (2006). *Robotic flowcharting with practical examples*. Robotics World.
- Corke, P. (2011). *Robotics, vision, and control: Fundamental algorithms*. Springer.
- Craig, J. J. (2005). *Introduction to robotics: Mechanics and control*. Pearson Prentice Hall.
- Flynn, A. M., & Jones, J. L. (1994). *Fast, cheap, and out of control: A robot invasion of the solar system*. MIT Press.
- Jones, J. L., & Flynn, A. M. (1993). *Mobile robots: Inspiration to implementation*. A K Peters.
- Katic, D., & Vukobratovic, M. (2011). *Intelligent control of robotic systems*. Springer.
- Khurmi, R. S., & Gupta, J. K. (2005). *Theory of machines*. S. Chand Publishing.
- Kumar, S., & Mittal, R. K. (2008). *Design of robotics and automation*. McGraw Hill.
- Osborne, A. (2012). Using Fischertechnik in engineering and robotics education. *TechPress Journal*.
- Parker, L. (2008). On the use of simulators for multi-robot systems research. *Robotics Journal*.
- Robomind. (2023). Robomind simulator documentation. Retrieved from www.robomind.net.
- Sabril, A., & Abdal, N. M. (2020). Comparison of mobile robot's travel time in the maze arena with left hand and right hand algorithms. *Jurnal Media Elektrik*, 17(3), 97–101.
- Siciliano, B., & Khatib, O. (2008). *Handbook of robotics*. Springer.
- Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: Modelling, planning, and control*. Springer.
- Siegwart, R., & Nourbakhsh, I. R. (2004). *Introduction to autonomous mobile robots*. MIT Press.