Simulation of Autonomous Driving for a Line-Following Robotic Vehicle: Determining the Optimal Manoeuvring Mode

Murat Bakirci
Unmanned/Intelligent Systems Lab, Faculty of Aeronautics and Astronautics, Tarsus University,
Mersin 33400, Turkey
muratbakirci@tarsus.edu.tr

***Abstract*—Mobile robotic systems offer valuable test platforms due to their shared features with autonomous vehicles, including features such as sensor technologies, navigation algorithms, and control systems. However, constraints in laboratory environments or technical resources, along with the need for extensive testing, often necessitate the use of virtual test laboratories. While line-following is a widely preferred application in mobile robotics, research on this topic within virtual laboratories is limited. This study pioneers the use of a car-like robotic vehicle in conducting line-following tests within a virtual laboratory environment. To facilitate these tests, a virtual simulator was developed to meet the requirements of realistic simulations. This simulator includes simulated elements, such as roads and environmental features, along with virtual sensors designed to collect and process dynamic motion data. An exceptional aspect of this study is its ability to collect consistent dynamic travel data by sampling realistic sensor information within a virtual environment. The developed line-following algorithm employs a controller to minimise lateral deviation while the robotic vehicle follows a road line during its movement. The study conducted virtual driving tests using two different manoeuvre modes on four distinct road segments, exploring how the manoeuvring style influences the driving quality. It was demonstrated that in the low manoeuvre mode, the ride is more comfortable, but exhibits a greater route deviation due to reduced oscillation, while the high manoeuvre mode exhibits the opposite behaviour.**

***Index Terms*—Autonomous drive; Virtual laboratory; Mobile robot; Manoeuvre analysis; Driving simulation; Line following.**

I. INTRODUCTION

Mobile robots are frequently used in various research areas due to their superior mobility, ease of operation, and relatively low cost [1]–[6]. These robotic vehicles find applications in education [7], [8], research [9]–[11], surveillance [12], industry [13], and military purposes [14]. Moreover, while more advanced hardware and software examples of these robots are used in space research [15], [16], their potential applications continue to grow [17], making them increasingly indispensable. In particular, wheeled mobile robots have seen substantial improvements in performance and capabilities, thanks to advances in both software and hardware. These advances have prompted the scientific community to embark on more comprehensive studies to maximise the efficiency of these robots [18]. Among the various types of mobile robotic vehicles, the most preferred are differential-drive mobile robots. Their simple architecture and well-established kinematic and dynamic models make them the best choice [19]. As the name implies, differential-drive robotic vehicles feature independent right and left wheels that share a common axis, as well as other wheels responsible for stabilising the robotic platform [20].

In addition to established kinematic and dynamic models, system identification methods are commonly used to analyse these robotic vehicles [21], [22]. Among the various approaches, the identification of data-driven systems based on input and output data is particularly favoured [23], as it is often straightforward to implement in laboratory tests [24]. Notably, this method is not confined to mobile robots; it is widely applied in the analysis of various other autonomous systems [25].

Line-following is one of the common research areas for testing mobile robotic vehicles [26]. These robots are typically equipped with a built-in wired control circuit and infrared (IR) sensors for precise line-tracking [27]. In line-following applications, a proportional controller (P) is commonly utilised, as discussed by the authors in [28]. The deviation from the desired path is determined by the distance between the IR sensor unit on the line and the central IR sensor unit. Many studies focus on enhancing the proportional controller by establishing a linear model of the robotic system. Maintaining a consistent line-following algorithm and a well-defined path for the robot to follow significantly enhances the functionality of these robotic vehicles. This approach enables full automation of numerous applications, thereby improving their overall efficiency.

The ability to independently control the motors of differential-drive mobile robotic vehicles gives them exceptional manoeuvrability, greatly improving their functionality [29]. The similarity between these robotic vehicles and conventional automobiles becomes more pronounced with the rapid advancements in MEMS sensors and automation technology, which have led to increased automation in road vehicles in the past decade. Today, nearly all mobile robotic vehicles have common features with conventional cars, with the primary distinction being the

Manuscript received 30 July, 2023; accepted 6 November, 2023.

absence of an internal combustion engine. Considering the ongoing advancements in electric vehicle technology, even this distinction is likely to diminish over time. If researchers in the field of mobile robotic vehicles take this trend into account, it could potentially herald a new era in transportation technology.

Simultaneously, the field of intelligent transportation systems (ITS) is progressing rapidly and presents new demands due to these advances. These needs encompass issues such as autonomous control [30], optimal utilisation of IoT [31], and immediate access to transportation data [32]. Incorporation of a wide array of sensors, microprocessors, and embedded measurement and control mechanisms into vehicles and transportation infrastructure underscores the rapid convergence of these seemingly distinct technologies. When applying the insights garnered from mobile robotic vehicle research to ITS, numerous transportation challenges can be resolved more efficiently and quickly. For example, emerging ITS applications can be initially tested using robotic vehicles, saving time and cost while minimising potential accident risks. This dual approach could establish precisely controlled line-following robotic vehicles as a pivotal component of the future of intelligent transportation. These vehicles serve as an invaluable platform for testing, validating, and enhancing ITS. Their controlled operating environment, cost-effectiveness, and safety advantages position them as a cornerstone in the development of future transportation systems that are safer, more efficient, and more intelligent.

Establishing the required line trajectory for line-following robot applications often requires a spacious laboratory environment. Moreover, setting up multiple tracks with diverse features simultaneously within the same laboratory is usually impractical. A highly effective solution is to simulate line-following experiments through the creation of an interactive virtual laboratory on a computer. An offline virtual laboratory is a powerful and valuable tool for practice and further experimentation [33]. While resources for line-following mobile robotic vehicles are abundant, the literature shows a relative scarcity of simulation studies in this field. To address this gap, this study primarily caters to this need by examining the line-following capabilities of a car-like robotic vehicle within a virtual laboratory. Developed using the Unity’s game engine, this virtual laboratory accounts for the potential integration of mobile robotic vehicles into ITS in the future and investigates the consistency of the virtual robotic vehicle in line-following across various types of roads.

II. SIMULATION ENVIRONMENT

A modelling and simulation environment was developed to simulate various driving tests. This software serves as a virtual testing simulator that can be configured to mimic real-world conditions. It allows the creation of a wide range of features within the virtual test environment, such as road lanes, buildings, various objects, and obstacles. It also enables the collection and processing of data using virtual sensors that replicate the behaviour of real sensors embedded in a virtual car-like robot (VCLR). Furthermore, there is the capability to create numerous virtual robots to act as testbeds to simulate more complex transportation scenarios. The controller code developed for a specific application operates seamlessly on both physical robots with similar features and virtual robots. The VCLR and the entire simulation domain are managed by a configuration code organised in XML that oversees the number of robots, their properties, and the types and criteria of all objects within the simulation environment. Depending on the type of test conducted, a variety of objects of different types and sizes can be placed in relevant parts of the simulation environment. In this particular study, where the VCLR is required to follow a designated line trajectory on the road, the ground properties of the virtual environment can be adjusted accordingly. Users have the flexibility to define the geometric properties of these lines to be followed and make any necessary modifications. Simulation consists mainly of three main submodules: visualisation, VCLR kinematics, and sensor sampling. The details of each submodule are provided in the following section.

A. Visualisation

Although simulation platforms like CoppeliaSim or Gazebo are well-suited for the type of work presented here, this study specifically favoured Unity. Each of these simulation platforms is popular and comes with its own distinct advantages and use cases, with selection largely based on the specific requirements of a given project. The primary rationale for opting for Unity in this context is its exceptional ability to create visually immersive and realistic 3D environments. Unity boasts powerful rendering capabilities and supports high-quality graphics, making it particularly suitable for projects where visual fidelity plays a crucial role. This attribute proves invaluable in scenarios that require a lifelike visual representation, such as training simulations or applications in the field of augmented reality. Moreover, Unity is equipped with robust physics simulation capabilities, an indispensable feature for modelling the intricate dynamics of robot vehicles, encompassing their interactions with the environment, various objects, and other entities. Furthermore, Unity’s extensive Asset Store features a wide array of pre-built 3D models, environments, and assets, significantly expediting the development of simulations. This vast library can substantially streamline the process and contribute to the creation of realistic virtual environments, ultimately saving both time and effort. One particular advantage of Unity is its capability to collect dynamic data through consistent sensor simulation. The visualisation engine initiates the simulation by reading the properties of the simulation environment and the file that contains all user-defined system input objects and specific configurations. Then it advances all definitions in the input file at user-defined time intervals, typically set at 20 ms. Through the developed user interface, users can interact with options such as changing the view mode during the simulation. They have the flexibility to observe the simulation from within the VCLR, essentially experiencing it from the driver’s perspective, or as an observer from a fixed point outside the VCLR. A screenshot of the simulation interface in top view is presented in Fig. 1. Additionally, essential metrics such as the distance covered during the driving test can be displayed and monitored during the simulation.

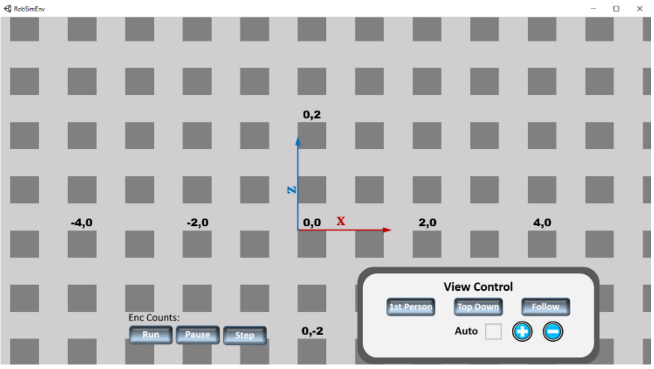

Fig. 1. Top-down view of the simulation environment launched in plain mode.

B. Kinematic Model of the VCLR

Modelling of the VCLR can be performed based on the basis of a rigid body balanced by wheels. With the flat-earth assumption, the VCLR body is assumed to be completely parallel to the ground, so it moves in a horizontal plane. VCLR (x_R, y_R) and inertial (x_I, y_I) coordinate frames are defined as shown in Fig. 2(a). The instant state of the VCLR in the inertial frame is represented by $\Gamma_I = [x \quad y \quad \theta]^T$, with x and y being the position and θ being the orientation. The instant state is defined by $\dot{\Gamma}_R = R(\theta)\dot{\Gamma}_I$ in the local VCLR framework, where R is the rotation matrix. From Fig. 2(b), where d is the width of the chassis of the VCLR, the right and left wheel velocities can be expressed as $v_r = \omega(r + d/2)$ and $v_l = \omega(r - d/2)$, respectively, where ICR denotes the instantaneous centre of rotation. With a trigonometric approach, the motion of the VCLR can also be expressed as follows:

$$\dot{x} = v\cos(\theta), \tag{1}$$

$$\dot{y} = v\sin(\theta), \tag{2}$$

$$\dot{\theta} = \omega, \tag{3}$$

where ω is the angular velocity and is expressed by $(v_r - v_l)/d$. v is the velocity of the geometry centre of the VCLR and is expressed by $(v_r + v_l)/2$. By integrating and rearranging (1)–(3), the following equations of motion are obtained:

$$x(t) = \int_{t_0}^{t} \left({V_R + V_L}/{2} \right) \cos\left[\theta(t)\right] dt, \tag{4}$$

$$y(t) = \int_{t_0}^{t} \left({V_R + V_L}/{2} \right) \sin\left[\theta(t)\right] dt, \tag{5}$$

$$\theta(t) = \int_{t_0}^{t} \left({V_R - V_L}/{w} \right) dt. \tag{6}$$

Like with sensor sampling, the game engine processes a series of samples at a specified sampling rate. Therefore, it is more advantageous to use the discrete form below

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t+1} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t} + \begin{bmatrix} dt\dfrac{V_L + V_R}{2}\cos\left(\theta_t + \dfrac{dt\omega_t}{2}\right) \\ dt\dfrac{V_L + V_R}{2}\sin\left(\theta_t + \dfrac{dt\omega_t}{2}\right) \\ dt\omega_t \end{bmatrix}. \tag{7}$$

Thus, by sampling the VCLR wheel velocities at a fixed time interval, Δt, the time-varying position is obtained with (7) in the inertial frame.

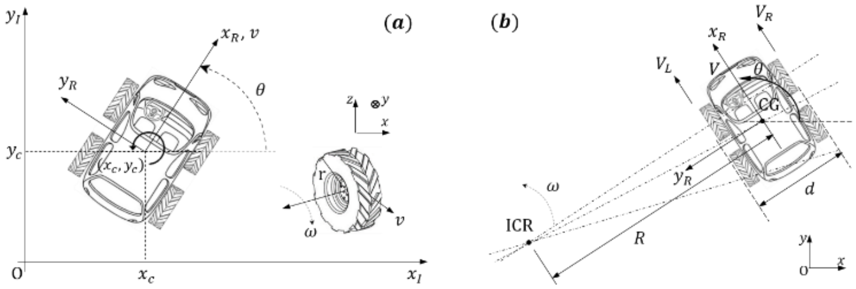

Fig. 2. (a) Inertial coordinate system and VCLR's local coordinate system; (b) Illustration of differential-drive feature of the VCLR.

C. Sensor Sampling

The simulation of the embedded robot sensors is achieved through the use of submodules, each dedicated to a specific type of sensor. In a virtual robot with predetermined physical and geometric characteristics, virtual sensors can be added or removed as needed. The selection of these sensors is dependent upon the requirements of the driving test, ensuring that they are seamlessly integrated into the robot to accurately collect data during testing. The road segments designated for testing are depicted in Fig. 3. These segments represent typical road scenarios that are often favoured in studies related to intelligent transportation systems.

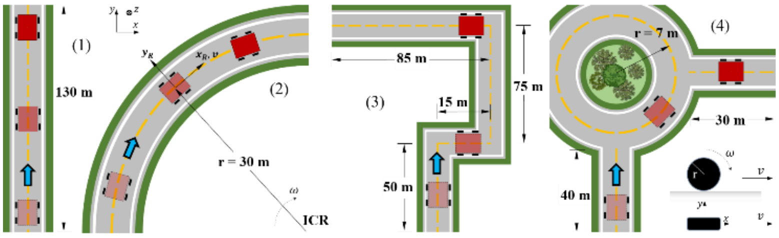

Fig. 3. Four different routes used in the simulation.

For this particular study, the VCLR is tasked with following a line trajectory, and to fulfil this function, a set of infrared (IR) sensors is defined within the simulation. This virtual array of IR sensors is positioned at the front of the VCLR, as depicted in Fig. 4. The downward-facing IR sensor array comprises eight individual sensor units, each composed of an illuminator and a detector. These sensors operate on the basis of the detection of IR light emitted by the illuminator. When the detector successfully registers the returning IR light, it outputs a value of 1; conversely, it returns 0 if no detection occurs. For example, the sensor would return a value of 0 when the IR signal is directed at a dark spot on a light background as it cannot be reflected. In contrast, it would return 1 when the IR light is directed at a light spot, detectable by the detector. This approach employs eight sensors operating in this manner, positioned alongside each other to collectively send eight IR lights toward the ground simultaneously, providing information about the geometric location of the line to be followed. By selecting a brightness threshold value, the system ensures that the detector outputs 1 if the detected light exceeds this threshold, and 0 if it falls below it. This threshold-setting approach is better aligned with the emulating of real-world conditions compared to the straightforward black-and-white differentiation, as often used in experiments. In practical settings, it is not uncommon to encounter scenarios where black route lines on a white background may not be readily discernible. Hence, establishing this threshold value equips the IR sensor set to effectively differentiate between the ground and the line under diverse conditions, ensuring its reliability in various real-world scenarios.

D. Simulation of the VCLR Kinematics

The visualisation engine serves as the core component for simulating VCLR kinematics by employing a third-order electric motor model. The model takes as input the pulse width modulation (PWM) signal sent to the motors responsible for controlling the VCLR’s wheels, and its corresponding output provides the rotational speeds of these wheels. It is important to note that the model factors in the contribution of the entire robot’s weight to the motor response, as this model was derived during the open-loop VCLR operation. Given the differential-drive characteristics of the VCLR, the model is executed individually for each wheel throughout the simulation.

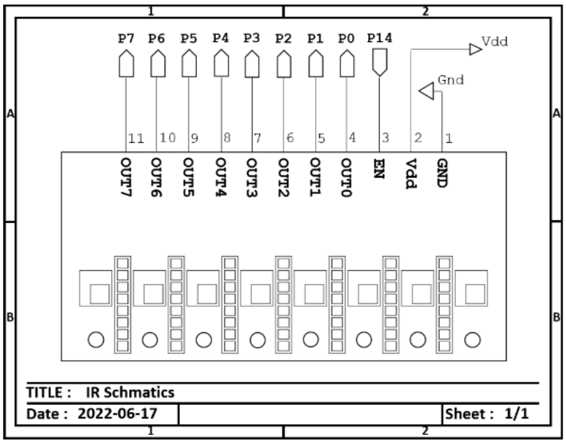

Fig. 4. Infrared (IR) sensor schematic.

E. Control of the VCLR

The developed line-following algorithm incorporates a controller aimed at minimising lateral deviation when the VCLR follows the road line during its motion. The offset, which signifies the distance from the line to be tracked, is determined by the IR sensor set and is conveyed as an eight-bit reading. Taking into account the simulation design and the thickness of the road line, a maximum of two IR sensor units can be placed on the line simultaneously. The eight-bit sensor reading is 00000001 when the VCLR is positioned at the far left of the line, and 10000000 when it is at the far right. In the ideal scenario, when the VCLR is perfectly centred on the line, the reading will be 00011000, indicating that the two IR sensor units in the middle are precisely on the line. This alignment, where the robot is perfectly centred on the line with zero error, serves as a reference, and any deviations from this value are recorded as errors. The controller processes this eight-bit real-time sensor reading to adjust the VCLR movement if it deviates from the line, thus maintaining its course.

III. Results

Before conducting tests in the virtual laboratory, a series of line-following experiments were performed in a real laboratory using two different robots. The purpose of these experiments was to demonstrate the correct functionality and applicability of line-following based on IR sensor readings with physical robots. Figure 5 illustrates the test robotic platforms and the experimental setup used. Both robots have a differential-drive feature, and their instantaneous position and heading information can be tracked using the developed kinematic model (7). They were equipped with the same set of IR sensors, consisting of eight IR sensors, mounted on the front of the robots, facing the ground. These robots were subjected to tests in a controlled environment where routes were defined using black electrical tape on a white cardboard background for line-following purposes. The reference line to be followed and the paths taken by both robots are depicted in Fig. 6.

As evident from the figure, both robots successfully completed the tests by following the line, albeit with some degree of deviation. Robot-1 exhibited an average heading error of 4.07 degrees, while Robot-2 had an average heading error of 2.51 degrees. On the basis of these results, it was anticipated that even lower errors would be observed in the virtual laboratory tests, as various unpredictable factors in the real environment, such as uneven terrain and wheel slippage, would not be present. Furthermore, taking into account current autonomous driving standards, lateral deviations of up to 0.2 metres and a maximum head error of 1.5 degrees are considered acceptable in the context of highway navigation [34]. Therefore, the evaluation of virtual laboratory tests would be based on these two criteria.

The algorithm designed for line-following governs the wheel speeds of the VCLR using the kinematic model to steer the robotic vehicle. In the initial simulation trial, three distinct steering commands are issued to the VCLR in response to three types of deviations detected by the IR sensor array. These steering commands are referred to as *mid-slow*, *mid-fast*, and *fast-sharp*, corresponding to the binary readings

00110000, 01100000, and 11000000, respectively. These binary readings indicate that the tracked line is to the left of the position of the robotic vehicle. In this scenario, the right wheel of the VCLR accelerates in response to the binary value, leading the robot to turn left. On the contrary, binary readings 00001100, 00000110, and 00000011 are used when the line is to the right of the centre of the VCLR. However, in each iteration, the algorithm initially checks if the line is already centred. If so, no steering commands are executed. If the line is not centred, it identifies the deviation from the line based on the binary reading from the IR sensor array and applies the appropriate steering command: mid-slow, mid-fast, or fast-sharp.

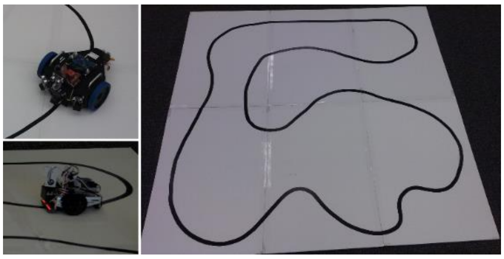

Fig. 5. Test robots and experimental line-following setup.

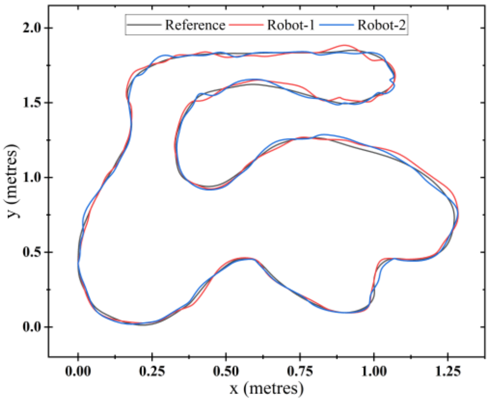

Fig. 6. Reference route and the trajectories followed by both robots on this route.

In the second simulation trial, five different steering commands are sent to the VCLR in response to five distinct deviations detected by the IR sensor array. These steering commands are labelled *slow*, *mid*, *mid-fast*, *fast*, and *sharp*, corresponding to binary readings 00010000, 00100000, 01100000, 01000000, and 10000000. Similarly, when the line is to the right, these readings are mirrored. The angular velocity profiles of the robotic vehicle corresponding to these different steering commands are presented in Fig. 7(a). An illustration of the angular velocity graph for both manoeuvring modes of the vehicle on Route-3 is displayed in Fig. 7(b). In this graph, positive angular velocity values denote a clockwise manoeuvre, while negative angular velocity values signify a counterclockwise manoeuvre.

Figure 8 illustrates the variation in the VCLR's heading as it traverses four distinct routes. It is important to note that in the graphs presented in Fig. 8, the x-axis represents the position, denoted as *r*, which indicates the distance from the initial starting point of the movement. Route-1, a straight road segment, provides an ideal scenario for investigation. To facilitate a more comprehensive analysis, the VCLR is initially placed with its IR sensor binary readings set to 00011000, indicating that it is centred on the line but orientated 35 degrees to the left (θ = 125 °) from its intended direction (θ = 90 °). This setup allows for observation of how the robot performs the necessary manoeuvres to follow the line, accounting for the significant errors present in the initial heading values for Route-1. On Route-1, operating in the 3-manoeuvre mode, the VCLR approaches the target line after covering approximately four metres.

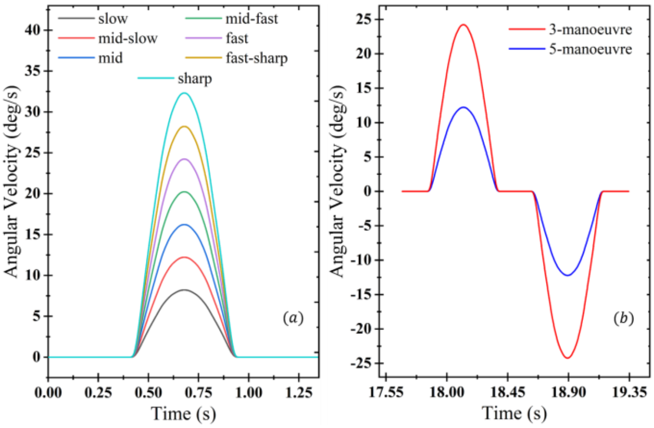

Fig. 7. (a) Angular velocity profiles of the robot corresponding to different steering commands; (b) An example of the angular velocity variation of both manoeuvre modes of the vehicle moving on Route-3.

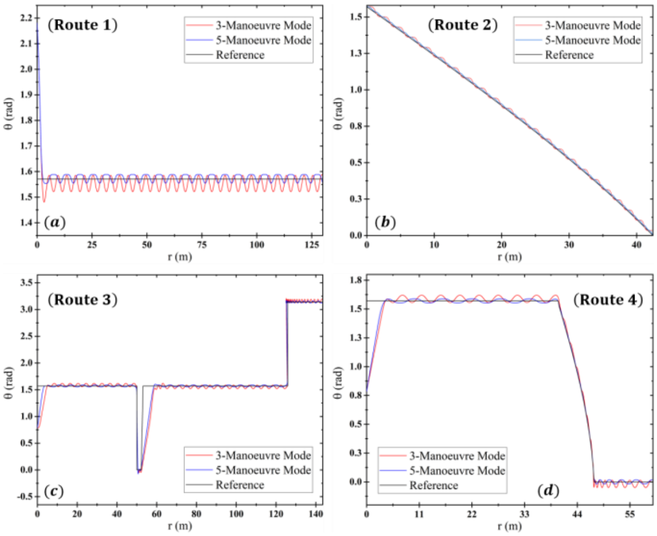

Fig. 8. Heading change according to the instant position of the VCLR tested on four different routes: (a) Route-1; (b) Route-2; (c) Route-3; (d) Route-4.

Beyond this point, it proceeds with slight oscillations around the line, eventually completing the route. The 5-manoeuvre mode, as expected, performs even better in Route-1. It gets very close to the target line after about two metres, benefiting from its enhanced manoeuvrability and exhibiting fewer oscillations compared to the 3-manoeuvre mode. The mean square error (MSE) values for both manoeuvre modes on each route are compiled in Table I. Route-2 involves the VCLR initially aligned with its intended direction (θ = 90 °). It successfully completes the route by making periodic slight rightward manoeuvres. Similarly to Route-1, the 5-manoeuvre mode outperforms the 3-manoeuvre mode, as can be seen in Table I. Route-3 commences with a straight segment, resembling Route-1. Consequently, the VCLR is initially placed to centre the line, but with an orientation of θ = 40 °. In the 3-manoeuvre mode, it aligns with the line approximately six metres into the movement, while the 5-manoeuvre mode achieves this in about three metres. The

VCLR then follows the straight segment with minor oscillations. At the end of this segment, the VCLR executes a full 90-degree right turn, which it accomplishes in both manoeuvre modes. Proceeding for 15 metres towards θ = 0 °, it makes a full left turn, similar to the first turn. In particular, the 5-manoeuvre mode consistently outperforms the 3-manoeuvre mode in these turns, as can be seen in Table I. The VCLR travels another 75 metres in the θ = 90 ° direction, concluding the route with another full turn to the left. Although it follows the line admirably along the straight segments, there is a slight delay in catching the line during turns. It should be noted that in each simulation trial, the robot's initial position is established as (0, 0). Consequently, the variation in *r* for horizontal movements appears less pronounced than for vertical movements. This explains why, although the VCLR performance appears similar in both turns, there is a difference in the representation of the graph. During these turns, it is evident that the 5-manoeuvre mode consistently outperforms the 3-manoeuvre mode. On Route-4, similar to the previous routes, the VCLR begins with an orientation of θ = 45 °, and, as expected, the 5-manoeuvre mode surpasses the 3-manoeuvre mode. On Route-2 and the circular section of Route-4, the VCLR delivered a performance even more remarkable than initially anticipated. It was projected to excel primarily on linear road segments, but it astoundingly completed these curved sections with minimal errors. Furthermore, taking into account the maximum heading error of 1.5 degrees (0.0262 radians), which is the autonomous driving standard mentioned at the beginning of this section, it is evident that the performance of VCLR falls within the acceptable limit for both manoeuvre modes on all routes, except for Route-3.

TABLE I. THE AVERAGE HEADING ERRORS.

	Mean Square Error (rad)	
Route No.	**3-Manoeuvre Mode**	**5-Manoeuvre Mode**
1	0.005662	0.004379
2	0.000459	0.000151
3	0.086536	0.051023
4	0.012396	0.008152

Figure 9 presents the total lateral deviation during the movement of the VCLR on each route. Across all four routes, the 5-manoeuvre mode consistently exhibits reduced lateral error. Routes 1, 3, and 4 exhibit significant initial lateral deviation errors, attributed to the initial orientation of the VCLR in varying directions. However, the robot rapidly rectifies these errors by implementing the required manoeuvres, quickly centring itself on the line as dictated by the algorithm. On Route-1, although the 5-manoeuvre mode appears to have minimal lateral deviation, the values of the y-axis do not adequately represent this error due to the initially substantial error. The distinction between the 3-manoeuvre and 5-manoeuvre modes becomes apparent in the second route, where the former results in roughly twice as many lateral deviation errors compared to the latter. Notably, the large deviations during full right and left turns in Route-3 are evident. In all three turns, the 5-manoeuvre mode executes more effective manoeuvres, swiftly rejoining the line, and incurring fewer errors. In the circular segment of Route-4, it is evident that the robot performs admirably in both manoeuvring modes, outperforming the rest of the route. Regarding the established performance criterion, the average lateral deviation errors on Route-1, Route-2, and Route-4 for both manoeuvre modes were found to be below the upper limit of the autonomous driving standard of 0.2 metres, as shown in Table II. In the case of Route-3, while the lateral deviation in the 3-manoeuvre mode slightly exceeds this threshold, it remains within the acceptable limit when using the 5-manoeuvre mode.

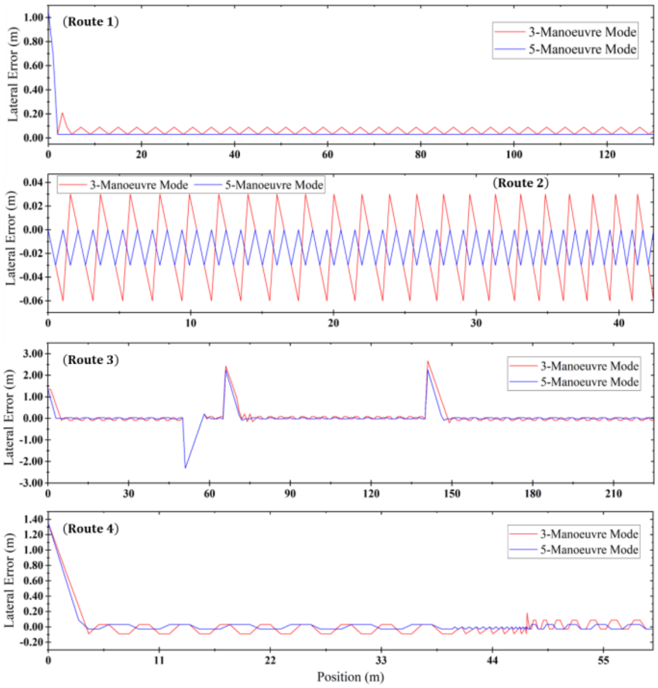

Fig. 9. Total amount of lateral deviation during the movement of the VCLR on each route.

TABLE II. THE AVERAGE LATERAL DEVIATION ERRORS.

	Mean Square Error (m)	
Route No.	**3-Manoeuvre Mode**	**5-Manoeuvre Mode**
1	0.07327	0.04282
2	0.02999	0.01483
3	0.20415	0.14176
4	0.08672	0.05286

Figure 10 provides insight into the speed differentials between the robot's wheels along each route, providing a glimpse of instantaneous manoeuvres as reflected in the corresponding speed variations.

It is important to note that the difference in speed between the wheels is directly related to the manoeuvrability of the robot. In essence, the ability to execute necessary manoeuvres with precision depends on how accurately the speed differential between the wheels is managed. When no manoeuvres are executed and both wheels rotate at the same speed, the speed differences are zero. Throughout the motion of the VCLR on all routes, the 5-manoeuvre mode aims to maintain a smoother line-following experience by minimising speed differentials. This is achieved through the design of the mode, which includes the selection of smaller control adjustments, enhancing its precision in manoeuvring. On Routes 1, 3, and 4, it is evident that the robot initiates more pronounced manoeuvres at the outset to quickly converge with the target line when employing the 5-manoeuvre mode. Similarly, as it approaches the line, it endeavours to remain in close proximity by executing more subtle manoeuvres.

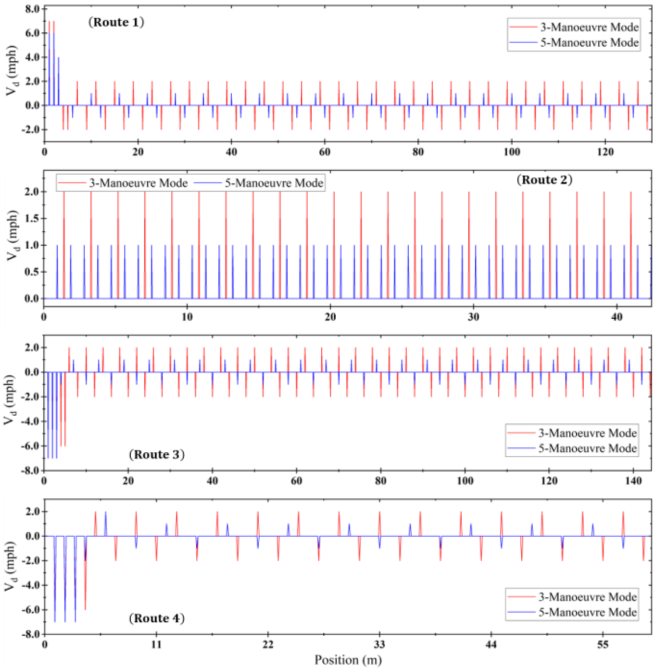

Fig. 10. Speed differences between the wheels of the robot along each route.

IV. CONCLUSIONS

Autonomous driving tests involving a car-like robot were simulated and studied within a virtual laboratory environment. The simulation environment, developed using the Unity game engine, underwent rigorous testing with various straightforward driving scenarios, confirming the high repeatability and reproducibility of the tests carried out. The results of these tests demonstrated that the developed kinematic model consistently represented the robot motion under optimal driving conditions. In simulations that included two distinct manoeuvre modes across a range of road types, the virtual car-like robot effectively navigated all phases, maintaining errors within the acceptable boundaries for autonomous driving, with only a few exceptional instances. A significant contribution of this study was the ability to sample realistic sensor data, which closely matched the sensor output obtained from real-world robot tests. This revealed that the virtual environment could support reliable and consistent testing. In cases where low manoeuvrability (3-manoeuvre mode) settings were applied to the virtual robotic vehicle, it displayed lateral deviations of up to 0.37 metres from the intended route. However, when set to high manoeuvrability (5-manoeuvre mode), it exhibited up to 65 % less lateral deviation. It is important to note that the high manoeuvrability mode, while more consistent in following the intended route, led to an increase in oscillation around the line due to the higher number of manoeuvres, which affected driving comfort. On the contrary, the low manoeuvrability mode offered a smoother overall ride experience. Although there was generally consistent performance in both driving modes, higher errors were observed during full turns, particularly on straight road segments or during circular motions.

CONFLICTS OF INTEREST

The author declares that he has no conflicts of interest.

REFERENCES

[1] M. B. Alatise and G. P. Hancke, “A review on challenges of autonomous mobile robot and sensor fusion methods”, *IEEE Access*, vol. 8, pp. 39830–39846, 2020. DOI: 10.1109/ACCESS.2020.2975643.

[2] H.-y. Zhang, W.-m. Lin, and A.-x. Chen, “Path planning for the mobile robot: A review”, *Symmetry*, vol. 10, no. 10, p. 450, 2018. DOI: 10.3390/sym10100450.

[3] J. Li, J. Wang, H. Peng, Y. Hu, and H. Su, “Fuzzy-torque approximation-enhanced sliding mode control for lateral stability of mobile robot”, *IEEE Trans. Syst., Man, Cyber.: Syst.*, vol. 52, no. 4, pp. 2491–2500, 2022. DOI: 10.1109/TSMC.2021.3050616.

[4] B. Liu, X. Xiao, and P. Stone, “A lifelong learning approach to mobile robot navigation”, *IEEE Robo. Autom. Lett.*, vol. 6, no. 2, pp. 1090–1096, 2021. DOI: 10.1109/LRA.2021.3056373.

[5] Q. Zhang, P. Wang, and Z. Chen, “An improved particle filter for mobile robot localization based on particle swarm optimization”, *Expert Syst. Appl.*, vol. 135, pp. 181–193, 2019. DOI: 10.1016/j.eswa.2019.06.006.

[6] M. Bakirci and B. Toptas, “Kinematics and autoregressive model analysis of a differential drive mobile robot”, in *Proc. of 2022 Int. Cong. Human-Comp. Interact. Optim. Robotic Appl.*, 2022, pp. 1–6. DOI: 10.1109/HORA55278.2022.9800071.

[7] M. J. Scott *et al.*, “Enhancing practice and achievement in introductory programming with a Robot Olympics”, *IEEE Trans. Educ.*, vol. 58, no. 4, pp. 249–254, 2015. DOI: 10.1109/TE.2014.2382567.

[8] D. Conti, S. Di Nuovo, S. Buono, and A. Di Nuovo, “Robots in education and care of children with developmental disabilities: A study on acceptance by experienced and future professionals”, *Int. J. Soci. Robo.*, vol. 9, pp. 51–62, 2017. DOI: 10.1007/s12369-016-0359-6.

[9] M. N. Zafar and J. C. Mohanta, “Methodology for path planning and optimization of mobile robots: A review”, *Procedia Comp. Sci.*, vol. 133, pp. 141–152, 2018. DOI: 10.1016/j.procs.2018.07.018.

[10] S. K. Malu and J. Majumdar, “Kinematics, localization and control of differential drive mobile robot”, *Glob. J. Res. Eng.: H Robo. & Nano-Tech*, vol. 14, no. 1, pp. 1–7, 2014.

[11] M. Bakirci and M. Cetin, “Improving position-time trajectory accuracy in vehicle stop-and-go scenarios by using a mobile robot as a testbed”, *Contr. Eng. Appl. Informatics*, vol. 25, no. 3, pp. 35–44, 2023. DOI: 10.61416/ceai.v25i3.8365.

[12] A. Sh. M. Al-Obaidi, A. Al-Qassar, A. R. Nasser, A. Alkhayyat, A. J. Humaidi, and I. K. Ibraheem, “Embedded design and implementation of mobile robot for surveillance applications”, *Indonesian J. Sci. Tech.*, vol. 6, no. 2, pp. 427–440, 2021. DOI: 10.17509/ijost.v6i2.36275.

[13] A. G. C. Gonzales, M. V. S. Alves, G. S. Viana, L. K. Carvalho, and J. C. Basilio, “Supervisory control-based navigation architecture: A new framework for autonomous robots in industry 4.0 environments”, *IEEE Trans. Indust. Info.*, vol. 14, no. 4, pp. 1732–1743, 2018. DOI: 10.1109/TII.2017.2788079.

[14] T. Guo, J. Guo, B. Huang, and H. Peng, “Power consumption of tracked and wheeled small mobile robots on deformable terrains–model and experimental validation”, *Mechanism Machi. Theo.*, vol. 133, pp. 347–364, 2019. DOI: 10.1016/j.mechmachtheory.2018.12.001.

[15] W. Gu, S. Cai, Y. Hu, H. Zhang, and H. Chen, “Trajectory planning and tracking control of a ground mobile robot: A reconstruction approach towards space vehicle”, *ISA Trans.*, vol. 87, pp. 116–128, 2019. DOI: 10.1016/j.isatra.2018.11.019.

[16] K. Di *et al.*, “Progresses and prospects of environment perception and navigation for deep space exploration rovers”, *Acta Geodaetica et Cartographica Sinica*, vol. 50, no. 11, pp. 1457–1468, 2021. DOI: 10.11947/j.AGCS.2021.20210290.

[17] J. Mu, X.-G. Yan, S. K. Spurgeon, and Z. Mao, “Nonlinear sliding mode control of a two-wheeled mobile robot system”, *Int. J. Model., Identif. Cont.*, vol. 27, no. 2, pp. 75–83, 2017. DOI: 10.1504/IJMIC.2017.082943.

[18] J.-y. Zhai and Z.-b. Song, “Adaptive sliding mode trajectory tracking control for wheeled mobile robots”, *Int. J. Cont.*, vol. 92, no. 10, pp. 2255–2262, 2019. DOI: 10.1080/00207179.2018.1436194.

[19] A. N. Amudhan, P. Sakthivel, A. P. Sudheer, and T. K. S. Kumar, “Design of controllers for omnidirectional robot based on the system identification technique for trajectory tracking”, *J. Phy.: Conf. Series*, vol. 1240, art. 012146, pp. 1–9, 2019. DOI: 10.1088/1742-6596/1240/1/012146.

[20] T. Q. Khai, Y.-J. Ryoo, W.-R. Gill, and D.-Y. Im, “Design of kinematic controller based on parameter tuning by fuzzy inference system for trajectory tracking of differential-drive mobile robot”, *Int. J. Fuzzy Syst.*, vol. 22, pp. 1972–1978, 2020. DOI: 10.1007/s40815-020-00842-

9.

[21] A. Mohamed, J. Ren, M. El-Gindy, H. Lang, and A. N. Ouda, "Literature survey for autonomous vehicles: Sensor fusion, computer vision, system identification and fault tolerance", *Int. J. Autom. Cont.*, vol. 12, no. 4, pp. 555–581, 2018. DOI: 10.1504/IJAAC.2018.095104.

[22] M. Bakirci, "Data-driven system identification of a modified differential drive mobile robot through on-plane motion tests", *Electrica,* vol. 23, pp. 619–633, 2023. DOI: 10.5152/electrica.2023.22164.

[23] E. Olcay, C. Dengler, and B. Lohmann, "Data-driven system identification of an innovation community model", *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1269–1274, 2018. DOI: 10.1016/j.ifacol.2018.08.358.

[24] C. Urrea and J. Pascal, "Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot", *Comp. Elect. Eng.*, vol. 67, pp. 791–806, 2018. DOI: 10.1016/j.compeleceng.2016.09.004.

[25] Y. Huang *et al.*, "An application of evolutionary system identification algorithm in modelling of energy production system", *Measurement*, vol. 114, pp. 122–131, 2018. DOI: 10.1016/j.measurement.2017.09.009.

[26] S. Suwasono, D. Prihanto, I. D. Wahyono, and A. Nafalski, "Virtual laboratory for line follower robot competition", *Int. J. of Electr. & Compu. Eng.*, vol. 7, no. 4, pp. 2253–2260, 2017. DOI: 10.11591/ijece.v7i3.pp2253-2260.

[27] E. S. Perez and F. J. Lopez, "An ultra-low cost line follower robot as educational tool for teaching programming and circuit's foundations", *Comp. Appl. Eng. Educ.*, vol. 27, no. 2, pp. 288–302, 2019. DOI: 10.1002/cae.22074.

[28] S. Saadatmand, S. Azizi, M. Kavousi, and D. Wunsch, "Autonomous control of a line follower robot using a Q-learning controller", in *Proc. of 2020 10th Annual Comp. & Commun. Workshop & Conf.*, 2020, pp. 0556–0561. DOI: 10.1109/CCWC47524.2020.9031160.

[29] A. Corrales Paredes, M. Malfaz, and M. Salichs, "Signage system for the navigation of autonomous robots in indoor environments", *IEEE Trans. Ind. Inf.*, vol. 10, no. 1, pp. 680–688, 2014. DOI: 10.1109/TII.2013.2246173.

[30] A. Ferdowsi, U. Challita, and W. Saad, "Deep learning for reliable mobile edge analytics in intelligent transportation systems: An overview", *IEEE Veh. Tech. Mag.*, vol. 14, no. 1, pp. 62–70, 2019. DOI: 10.1109/MVT.2018.2883777.

[31] A. Sumalee and H. W. Ho, "Smarter and more connected: Future intelligent transportation system", *IATSS Res.*, vol. 42, no. 2, pp. 67–71, 2018. DOI: 10.1016/j.iatssr.2018.05.005.

[32] F. Zhu, Y. Lv, Y. Chen, X. Wang, G. Xiong, and F.-Y. Wang, "Parallel transportation systems: Toward IoT-enabled smart urban traffic control and management", *IEEE Trans. Int. Trans. Syst.*, vol. 21, no. 10, pp. 4063–4071, 2020. DOI: 10.1109/TITS.2019.2934991.

[33] S. Tselegkaridis and T. Sapounidis, "Simulators in educational robotics: A review", *Educa. Sci.*, vol. 11, no. 1, p. 11, 2021. DOI: 10.3390/educsci11010011.

[34] T. G. R. Reid *et al.*, "Localization requirements for autonomous vehicles", *SAE Int. J. of CAV,* vol. 2, no. 3, pp. 173–190, 2019. DOI: 10.4271/12-02-03-0012.