International Journal of Multiphysics
Volume 18, No. 3, 2024
ISSN: 1750-9548

Improved Mobile Robot Manoeuvring Using Bayes Filter Algorithm Within the Planned Path

Mohammed Saad[1], Yarub Alazzawi[2]

[1] Department of Mechatronics Engineering/ Al-Khwarizmi College of Engineering/ University of Baghdad, Iraq

[2] Department of Mechatronics Engineering/ Al-Khwarizmi College of Engineering/ University of Baghdad, Iraq

mohammed.njm1903@kecbu.uobaghdad.edu.iq

ABSTRACT

This research introduces a novel approach for object tracking, capitalizing on the Bayes filter algorithm within the constraints of a single-camera setup. Object tracking is a pivotal aspect of computer vision, significantly influencing system performance in diverse applications. The integration of the Bayes filter algorithm provides a probabilistic framework, effectively addressing challenges posed by occlusions, lighting variations, and unpredictable object movements in real-world scenarios. Our methodology not only streamlines the tracking setup by utilizing a single camera but also enhances practicality, making it particularly relevant for applications with resource constraints. The paper offers a comprehensive exploration of this approach, delving into the theoretical foundations and technical intricacies that underlie the fusion of advanced object tracking techniques with the Bayes filter algorithm. Through empirical evaluations across varied tracking scenarios, our approach demonstrates superior effectiveness compared to traditional methods, showcasing the algorithm's ingenuity in improving tracking accuracy and adaptability. It achieved a dynamic simulation efficiency of 97.025%, a sensitivity of 96.2616%, and an overall system quality (F-score) of 97.0493%. This research contributes valuable insights to the evolving landscape of object tracking methodologies, presenting a practical and efficient solution that combines the Bayes filter algorithm's power with the simplicity of a single camera setup. The findings presented herein offer a nuanced perspective for researchers and practitioners seeking to elevate the precision and real-time adaptability of object tracking systems in diverse applications.

INDEX TERMS Baysian Filters, Kalman Filter, Robot Car, Coordinates Transformations, Auto Vehicle Driving, Object Tracking, Visual Processing, Sliding Mode Controller (SMC).

I. INTRODUCTION

The pursuit of real-time and accurate object tracking has become increasingly vital in the field of computer vision, finding applications in surveillance, robotics, and human-computer interaction. This paper explores a good approach to object tracking, leveraging the capabilities of sensors and employing the Bayes filter algorithm. This approach seeks to address the challenges associated with tracking objects in dynamic environments, providing a robust and adaptive solution. Traditional object tracking methods often contend with issues such as occlusions, changes in lighting, and unpredictable object movements, necessitating innovative solutions to enhance tracking accuracy. The integration of the Bayes filter algorithm, a probabilistic framework renowned for its ability to model uncertainty and update predictions iteratively, presents a promising avenue to tackle these challenges and refine the precision of object tracking. In this paper, we present a elucidating the fundamental principles that underpin the fusion of advanced object tracking techniques with the Bayes filter algorithm [1-4]. The utilization of a single camera system adds a layer of practicality to our approach, making it applicable to scenarios where resource constraints or simplicity in setup are paramount. Through empirical evaluations on diverse tracking scenarios, we aim to demonstrate the effectiveness of our approach in comparison to traditional methods, showcasing the impact of the Bayes filter algorithm on tracking accuracy and adaptability. The ensuing sections will delve into the

International Journal of Multiphysics
Volume 18, No. 3, 2024
ISSN: 1750-9548

technical details of our method, shedding light on its theoretical foundations, implementation nuances, and practical implications. This endeavor contributes to the evolving landscape of object tracking methodologies, offering insights into the potential of utilizing the Bayes filter algorithm in conjunction with sensors for well and real-time tracking applications. Figure 1.1 shows a schematic diagram of mobile robotic car techniques [5-8].

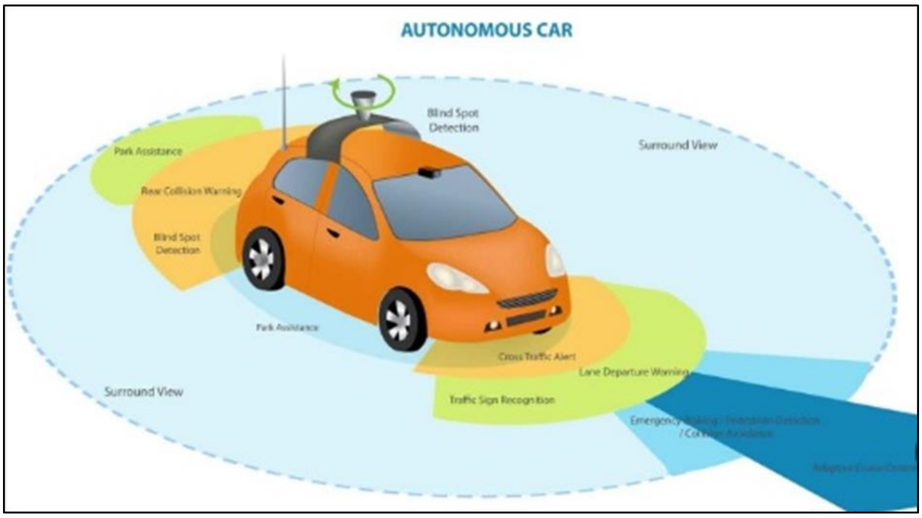


FIGURE 1. **Schematic diagram of self-driving car techniques [5-8]**

The conduct of drivers plays a crucial role in ensuring the efficient operation of vehicles on roadways. A considerable 94% of collisions and traffic incidents are attributable to reckless driving behaviors exhibited by drivers. To tackle the challenges associated with road accidents and vehicle crashes, Highly Automated Vehicle (HAV) Technologies have been suggested. Among these technologies, Self-Driving Cars stand out, offering significant advantages including enhanced road safety, increased autonomy, cost savings, heightened productivity, reduced traffic congestion, and environmental benefits.

A. Self-Driving Cars

Self-driving cars, or autonomous vehicles, operate without human intervention using sensors, cameras, and AI, they range from Level 0 (human control) to Level 5 (full automation), they promise safer roads, less congestion, and increased mobility. Companies like Google, Tesla, and Uber are developing this technology. Self-driving cars could revolutionize transportation, offering new possibilities for mobility and safety. Figure 2.4 displays the comprehensive technical structure of self-driving vehicles equipped with an autonomy system of Level 3 [15-18].

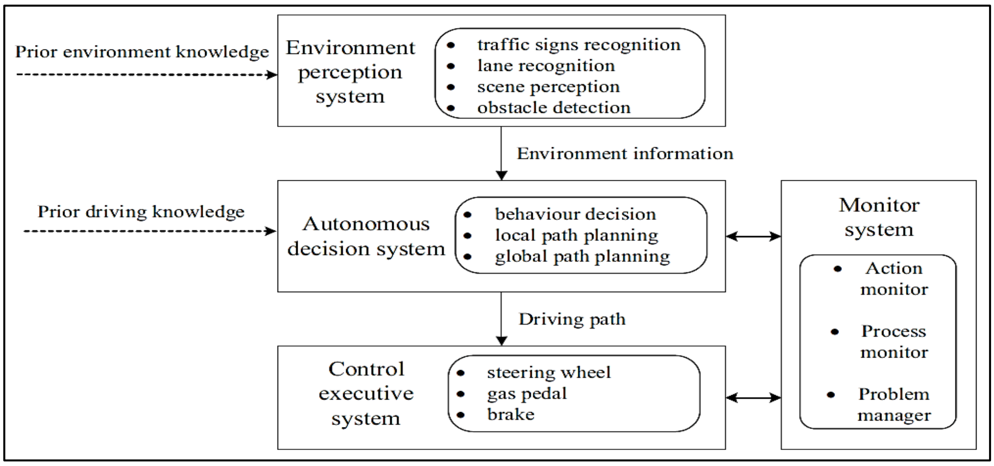


FIGURE 2. **System diagram for autonomous vehicles with Level 3 autonomy [15-18].**

Deep learning plays a crucial role in enabling self-driving cars by enhancing their perception, cognition, and decision-making abilities. Convolutional Neural Networks (CNNs) are used to detect objects such as pedestrians and vehicles in real-time from sensor data. Real-time processing is essential for quick responses to changing road conditions. Informed decisions are based on training models with large datasets, while adaptability allows for continuous improvement over time. Integration of sensor data from various sources like cameras, LiDAR, and GPS provides a comprehensive understanding of the driving environment [16, 19]. As autonomous driving technologies proliferate, they are expected to become prevalent in smart cities, with millions of these vehicles communicating with each other. This necessitates scalable, robust, secure, fault-tolerant, and interoperable technologies to support them effectively. Figure 3 displays a depiction of self-driving vehicles operating within smart urban environments [17-20].

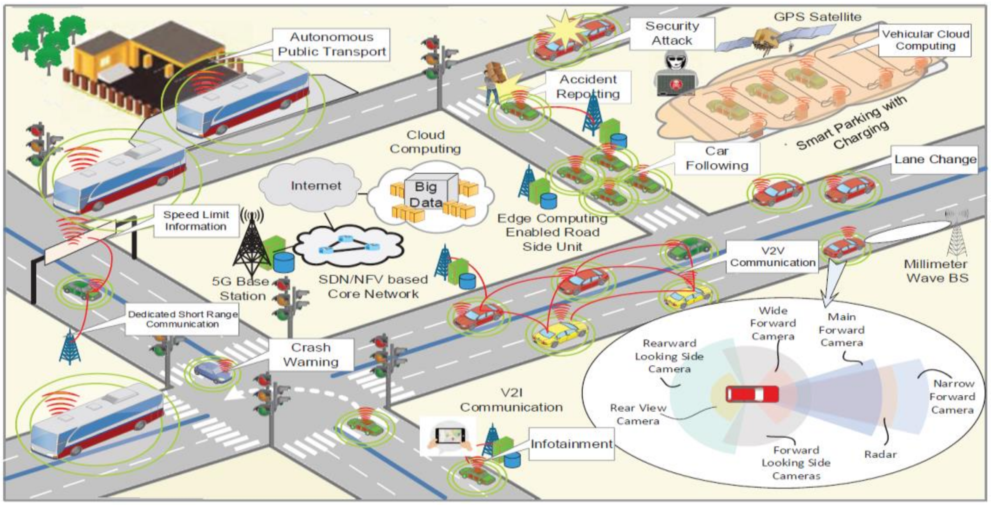


FIGURE 3. **A depiction of self-driving vehicles operating within smart urban environments [17-19].**

B. Mobile Robot

Mobile robots are autonomous systems that can navigate and operate in unpredictable and partially unknown environments without human intervention, they are designed to perform tasks such as carrying heavy objects, monitoring, search and rescue missions, and more, they rely on a range of environmental sensors to perceive and interact with their surroundings, these sensors can be mounted on the robot itself or positioned externally in the environment. The design of mobile robots includes considerations of locomotion, perception, and navigation, locomotion systems can vary depending on the terrain and the specific tasks the robot is designed for, with options such as wheeled, legged, walking, or hybrid designs. Navigation is a fundamental challenge in mobile robotics, requiring the robot to determine its current position, destination, and the optimal path to reach that destination. Techniques such as perception, localization, cognition, and motion control are employed to enable autonomous navigation. Figure 4 demonstrates the difficulties faced by mobile robots [18-22].

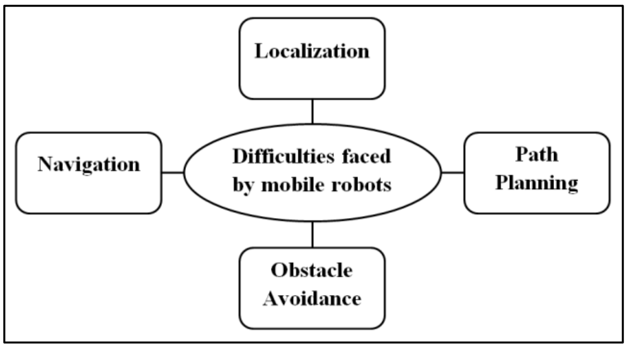


FIGURE 4. **Mobile robots difficulties structural diagram [18-22].**

The realm of robotics focuses on crafting and constructing systems capable of exhibiting a certain level of independent decision-making. Such systems, also referred to as cyber-physical systems, engage with the physical environment by amalgamating sensor inputs, actuators, and software algorithms. More specifically, it delves into Autonomous Mobile Robots (AMRs), which utilize wheels to traverse unaltered environments. This sets AMRs apart from Autonomous Guided Vehicles (AGVs), which rely on specialized infrastructure like sensor beacons or guiding strips for navigation [18-20].

C. The Bayesian Filter

Bayesian filtering calculates the posterior distribution of a state variable based on past measurements, crucial for estimation and tracking in signal processing and control systems. It involves recursive equations to compute predicted and filtering distributions at each time step, often using the Chapman-Kolmogorov equation for predictive distribution. This method accommodates linear and nonlinear systems without requiring explicit Gaussian assumptions for noise processes. Although useful for its flexibility, dealing with nonlinear models in virtual filtering is more complex compared to alternative techniques. Nonetheless, it remains a powerful tool for dynamic system analysis and tracking in various fields. One common application of Bayesian filtering is in tracking the position of a moving object based on noisy sensor measurements. It can estimate the current state by utilizing all measurement observations up to and including a specific time, making it a recursive filter, the filter uses importance sampling to choose weights for the samples and provides a weighted approximation of the posterior probability distribution. Bayesian filters, such as the Kalman filter, particle filter, and non-parametric information filter, have been used to improve the accuracy of proximity estimation in Bluetooth Low Energy beacon systems [19-26].

1) Role of Bayes' Theorem in Bayesian Filtering:

Bayes' theorem is the foundation of Bayesian filtering, providing a mathematical framework for updating probabilities based on new evidence. In Bayesian filtering, Bayes' theorem is used to calculate the posterior probability of a hypothesis given observed data, the theorem allows for the incorporation of prior knowledge or beliefs (prior probability) and the likelihood of observing the data given the hypothesis (likelihood) to compute the updated probability (posterior probability). By continuously updating the probabilities based on new evidence, Bayesian filtering can adapt and improve its accuracy over time [20-27]. Bayesian filtering equations are used to compute the marginal posterior distribution or filtering distribution of the state variable given a history of measurements, where equations involve the recursive computation of the predicted distribution and the filtering distribution at every time step. The prediction step of the predictive distribution of a state variable at a given time step is calculated using the Chapman-Kolmogorov equation [20-28].

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})\, p(x_{k-1}|y_{1:k-1})\, dx_{k-1} \quad (1)$$

The update step computes the posterior distribution of the state variable given the measurement at that time step using Bayes' rule [20-28].

$$p(x_k|y_{1:k}) = \frac{1}{Z_k} p(y_k|x_k)\, p(x_k|y_{1:k-1}) \quad (2)$$

The normalization constant is used to ensure that the posterior distribution is properly normalized [20-28].

$$Z_k = \int p(y_k|x_k)\, p(x_k|y_{1:k-1})\, dx_k \quad (3)$$

Also, Figure 5 demonstrates the representation of the update process [20-28].

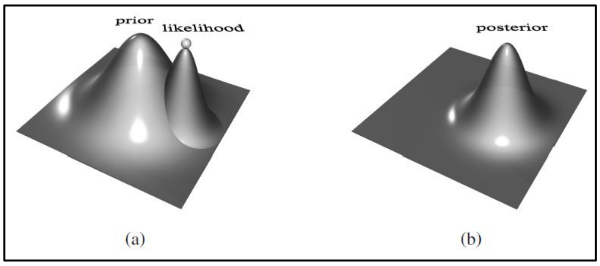


FIGURE 5. **The update process representation: (a) Initial distribution derived from prediction & the measurement likelihood prior to the update; (b) Resulting posterior distribution following the amalgamation of the initial and likelihood distributions using Bayes' theorem [30-34].**

Moreover, the Bayes filter is a probabilistic methodology employed to infer the condition of a system based on provided observations. It is frequently applied in the realms of robotics, autonomous vehicles, and other disciplines concerned with state estimation amidst uncertain conditions. Diverse approaches exist for conceptualizing and mathematically defining the Bayes filter. Figure 6 shows the Bayes Filter representation ways [22-30].

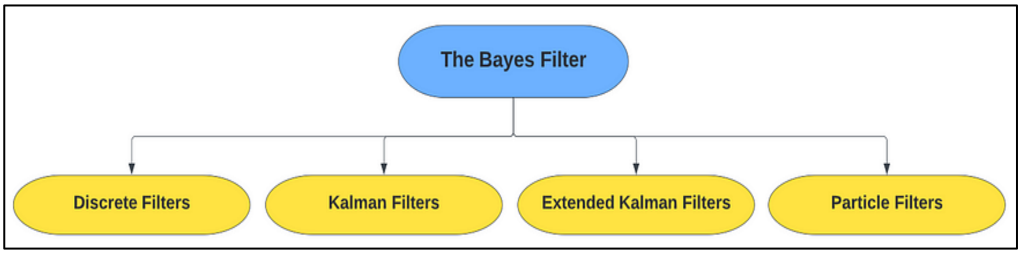


FIGURE 6. **The representation ways of Bayes Filter [22-30].**

2) Extended Kalman Filter (EKF)

The Kalman Filter, introduced by Rudolf Kalman since 1960s, is a prominent algorithm for estimating the hidden state of dynamic systems from noisy observations. Initially developed for linear state space models with additive Gaussian noise, it's renowned for its minimum mean-squared error estimation and applicability to discrete-time, time-varying systems. Operating recursively, it updates state estimates based on current observations and previous estimates, making it widely used in fields like radar tracking, trajectory estimation, and space navigation. Beyond its deterministic interpretation in minimizing least squares disturbances, the Kalman Filter is extensively employed in diverse domains like finance, aerospace, robotics, and signal processing. Its discrete-time equations facilitate state and error covariance estimation from noisy measurements, leveraging mathematical system models for accurate state estimation. Despite its prevalent usage, the filter's application isn't limited to Gaussian noise; it's a versatile tool for extracting valuable insights from noisy data. To enhance its adaptability, researchers integrate the Kalman Filter with neural networks, mitigating limitations posed by impractical system models and assumptions. By combining approximations, it refines state estimates, aiding in real-time control and estimation tasks across various stochastic systems. Thus, understanding the Kalman Filter's principles not only improves system control but also unlocks its potential in optimizing state estimation amidst diverse noise conditions, cementing its status as a foundational tool in information engineering and beyond [30-38]. The Extended Kalman Filter (EKF) is a specialized version of the Kalman Filter tailored for estimating the state of nonlinear dynamic systems, offering an expansion beyond the original Kalman Filter designed for linear systems, it achieves this by linearizing nonlinear system dynamics and measurement equations through a first-order Taylor series approximation, this enables the EKF to effectively handle nonlinearities within the system. Comprising prediction and update steps, the EKF first forecasts the current state estimate based on system dynamics and then refines it using available measurements. Widely applied across navigation, robotics, and control systems, the EKF serves as a reliable method for estimating the state of nonlinear systems, leveraging sensor fusion and error correction to enhance accuracy, specifically in mobile robotics, the EKF is instrumental in simultaneous localization and mapping (SLAM) tasks, enabling robots to navigate unknown environments and identify landmarks. By approximating nonlinear functions with linear ones, the EKF proves versatile and efficient in SLAM algorithms, utilizing the Kalman gain matrix to refine state estimates based on sensor data. Compared to its predecessor, the Kalman Filter, the EKF offers improved precision, stability, and efficiency, particularly in scenarios involving multiple landmarks. Its successful deployment spans various applications, including disaster response and autonomous robot operations, highlighting its significance as a powerful tool for accurately mapping environments and determining robot locations. Figure 7 demonstrates the extended Kalman filter block diagram [39-50].

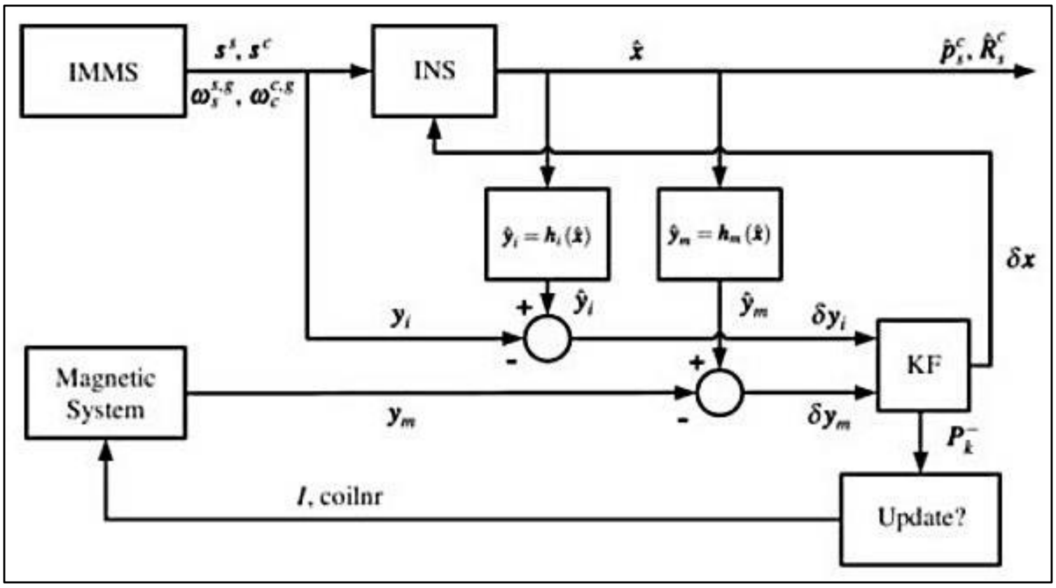


FIGURE 7. **Demonstrations of the extended Kalman filter block diagram [38-50].**

Also, particle Filter (PF) is a versatile data assimilation method with applications spanning geosciences, tracking, and state estimation in nonlinear systems, initially developed for high-dimensional geoscience systems, recent enhancements tackle inefficiencies, incorporating improved proposal densities, optimal transportation concepts, and adaptive resampling strategies, PF's success in atmospheric and oceanic applications suggests future prominence, potentially rivaling conventional methods in numerical weather prediction. In tracking, PF offers probabilistic estimation of uncertainty, accommodating diverse observation models and motion priors, critical for robust object pose tracking. As a statistical pattern recognition tool, PF utilizes non-parametric Monte Carlo simulation for accurate posterior state estimation through weighted particle representations. Its efficacy in nonlinear and non-Gaussian scenarios finds utility across diverse domains like target tracking, signal processing, and automation. In image segmentation, PF, integrated with deep learning, efficiently models observation distributions, enhancing proposal distribution accuracy by combining transition and observation models. Overall, PF emerges as a robust framework, adapting to complex systems and demonstrating potential for widespread adoption in various scientific and engineering disciplines [38-50].

II. Related Studies

Various studies and references connected with the issues of multi-reason mobile robots in view of microcontrollers and remote innovation have been gotten to in the references. In this segment, we will audit the latest distributions and articles connected with this title, additionally summing up them constantly of distribution. About getting a wide-ranging opinion as accessible with the most recent multi-reason mechanical upgrades with refreshes that address such points, as opposed to building a coordinated idea to foster the examination issue likewise to settle the targets and show up at the supportability of this survey, as well as to propose feasible outcomes with the examination permitted in this Survey. The most recent articles and exploration covering the functioning title are recorded underneath: In 2021, Vagale, A., et. al., [12], algorithms that don't need map portrayal (progressed) and those that do (exemplary) have been introduced as a charming division in robot development control. High level delicate registering with inspecting depended algorithms is taken care of by the customary method that contains diagram research draws near. In 2013, Souissi, O., et. al., [13], proposed a few self-evident and reasonable classes for route planning: depended on the robot model (complete, non-all out, kinematic); likewise founded on the guide model's requests (whether they were determined ahead of time); depending on the capacity to remap (disconnected or associated). Moreover, and under basic development criteria (deterministic or probabilistic), the calculations reliably indicate a generalization of a similar arrangement. In 2012, Jeong, S., et. al., [14], it has been exhibited and explored how surface mobile robots could go starting with one area in space then onto the next. Thus, the route planner ought to consider how the movement plan will cooperate with such a surface. A few algorithms, for example, require the making of a diagram that precisely portrays the setting with the end goal that the robot is working. Diagram Search algorithms, that fall underneath the C-Space search classification, ordinarily show this. The outline could likewise be used by transformative algorithms like Ant Colony Optimizers (ACO). This resource could address how the spatial course of action of the situation's route is impacted by territory features. In particular, the chart being referred to is proposed to be depended on measurement maps here, however another guides sorts, like topological and semantic guides, are past the extent of this review. In, 2013, Nash, A., et. al., [15], introduced a grouping of way maps that are better-figured out because of crafted by those specialists. Such exploration showed and made sense of that it recognizes guides and cell examination. The surface mosaic of the initial two cells is framed. Networks, whether standard or irregular, could be used to orchestrate such cells. In 2020, Bergman, K., et. al., [18], the movement of robots was analyzed using vision charts and cross section state diagrams. The last option includes causing Edges to depend on movement replacements, guaranteeing that the subsequent way is attainable because of bot development limitations, especially while utilizing chart search algorithms. Such diagrams' cells or hubs could store static or dynamic components of information about the surface at their positions. This may be, for example, determined as data concerning rise. In 2013, Papadakis, P. [19], examined a Computerized Rise Guide (DEM) lattice in which each center point has a height sum joined to it. Polygons could additionally be used to decide height maps, yet ordinary matrix maps are liked. Convolution matrices could be used to extricate features that are connected with shapes, similar to the slant or the roughness of the surface. What sorts of features are extricated not entirely set in stone by the portion size and DEM resolution. In addition, the degree of detail of the guide's not entirely settled by this resolution. This resolution could be either various or non-uniform. The scale at which the arranging is done may be utilized to choose the lattice's size: Worldwide assuming the region is bigger than that, commonly using data from outside sources like satellites or robots; nearby assuming the robot's prompt environmental factors (roughly the reachable length of the on-board sensors) are covered. In 2020, Effati, M., et. al., [20, 21], investigated the robot's movement setup as a subject of exploration. Coming up next are important for the different kinetic and dynamic models that the analysts checked on in this study which drived by differential. The base turning sweep of robots with Front Ackermann steering and the high energy utilization of slide steering robots through turning maneuvers are two instances of imperatives that are relevant to way arranging. In 2010, Patel, N., et. al., [22], an elective construction happened, named Crabbing. It allows a robot to move in an orientation variant to the heading it is noticing in light of the fact that it will give driving joints on the upper side of each and every mechanical wheel. In 2015, Brunner, M., et. al., [23], explored a few kinematic designs to allow the robot to give the Point Turn move, which makes them to spin past transmission. It is an advantage to relate the event of explained robots that are fit for inward plotting to acquire a piece of sort of advantage too give various sorts of locomotion. The explained robots, against tracks, for example, could proficiently control their dependability while steering on rough terrains. In 2019, Sánchez-Ibánez, J.R., et. al., [24], recognized the reconfiguration capacity of way arranging algorithms are really important for such robot kind, as they could find gets to that take advantage of their colossal versatility. In the issue of worldwide preparation, the creators used a Chart Search calculation, Dijkstra, to design an entrance and afterward, by reproduction hardware, process that locomotion made is ideal to move each piece of its segments. Subsequently, the creators of such review distribution proposed the execution of a PDE assessment technique to respect the multi-locomotion models using an isotropic expense capability at the instant of planning. In 2017, Norouzi, M., et. al., [25], proposed the execution of the Quick Walking Tree (FMT), the Inspecting Based calculation, to handle the inspiration planning of a re-plotted wheeled-legs hybrid robot. The locomotion of the robot will adjust best or more awful depending on the territory properties. Such attitudes may be reliant upon either the morphology (design) or the territory arrangement. One of such element is the slant or landscape tendency. The incline influences the robot's Roll and

Pitch orientation points, which is important to be respected for soundness safeguarding. In 2016, Barjuei, E.S., et. al., [26], An adaptable L-molded instrument with gravity recommended for orchestrating powerful controllers, in view of H¥ circle forming and tuning m for both position control and spatial vibration damping. In 2017, Rsetam, K., et. al., [27], a progressive non-peculiarity station SMC has recommended, which could guarantee a quicker intermingling rate to zero of framework states inside for a limited period and peculiarity free. In 2019, Shokoohinia, M.R., et. al., [28], a control regulation for eyewitness responses has recommended depending on the limited component approach with model minimization. Concerning robustness, among the control methods, the sliding mode controller (SMC) is a simple methodology, which delineates security against non-model boundary varieties outer dynamics as well as unsettling influences. In 2019, Soltanpour, M.R., et. al., [29], a voltage-based SMC is recommended, that has low computational volume. In spite of the intrinsic advantages of robustness contrasted with aggravations, the presentation of SMCs may be impacted by changing framework boundaries. In 2020, Mahmoodabadi, M., et. al., [30], used a way to deal with change nonlinear dynamics towards linear dynamics; hence, a sliding position control technique has executed as a way following controller. Strange model, is utilized to control the increase of the control center. In 2019, Zaare, S., et. al., [31], a voltage-subordinate sliding mode is considered where the controller is joined with a versatile assessor. The control unit that could be manipulated and work the vulnerabilities on the heap side and the drive side at the specific time frame is as yet absent. In 2021, Tuan, H.M., et. al., [32], executed the controller on a two DOFs robot arm with adaptable actuators. What's more, a versatile control is portrayed that expects to settle the model and furthermore manage vulnerabilities for the impact of outside force/torque not delivered. In 2021, Hao, N.V., et. al., [33], extended the examination by giving further reproduction tries and proposing new algorithms to further develop the control cycle. A correlation was formed with a conventional sliding mode controller to show the viability of the proposed calculation.

III. Methodology

In this study, the suggested mobile robot-based target follower using bayes filter algorithm model has been considered and designed. In fact, Mobile Robot-based target follower using Bayes Filter algorithm is an important topic of autonomous vehicle navigation. By incorporating such technology, vehicles could efficiently also safely navigate complicated highway environments, regarding parameters such as velocity, acceleration, as well as lateral motion.

A. Model Design

In order to construct the structure of the Mobile Robot-based target follower using Bayes Filter algorithm model, Figure 8 illustrates a block diagram of general methodology through the utilization of Bayesian filtering for autonomous driving and navigation.

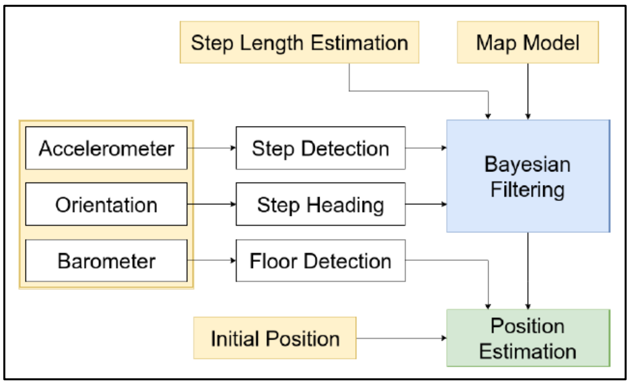


FIGURE 8. Block diagram of the method of working through the use of Bayesian filtering on the topic of autonomous driving and objects tracking.

The aim of this research is to evaluate and compare various Bayesian filtering approaches those probabilistically structures the uncertainty presented by noisy estimations. Despite of optimal techniques are not chosen for every particular problem. Also, their responce is investigated inaccurate factors lead to the complete solution setting, for example, how strategies are configured dealing with underestimated or overestimated step lengths. An overview of our technique is described in Figure 9 with the chosen approaches comprehension.

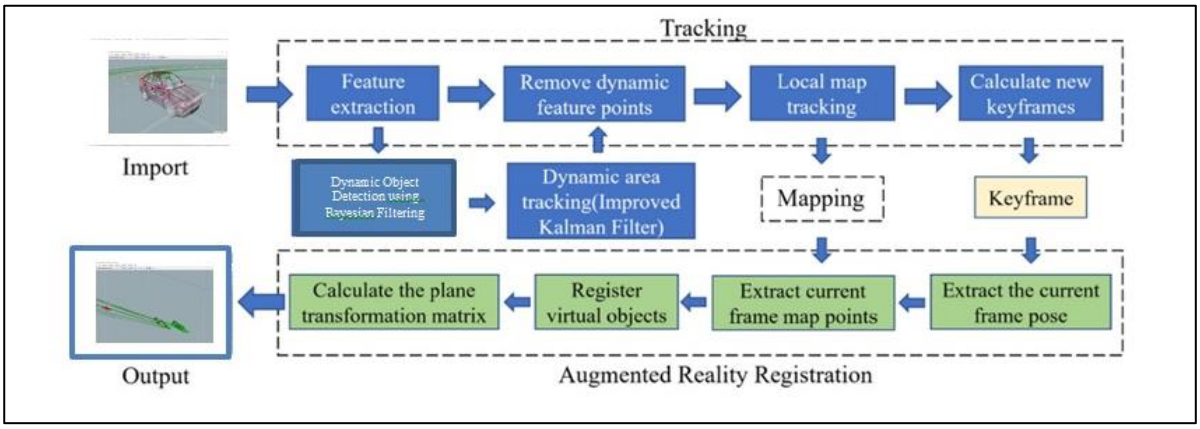


FIGURE 9. **Block diagram of auto driving model based on object detection in dynamic environments.**

The above diagram shows the autonomous driving system in terms of dynamic target detection using traditional methods, which is greatly affected by variations in scene brightness, noise, etc. There will be false discoveries and missed discoveries in the target through the detection operation. This further leads to drift of results along the target tracking, which in turn changes target tracking accuracy. Thus, Figure 10 displays the MATLAB R2018a Simulink block diagram of the robotic vehicle suggested model.

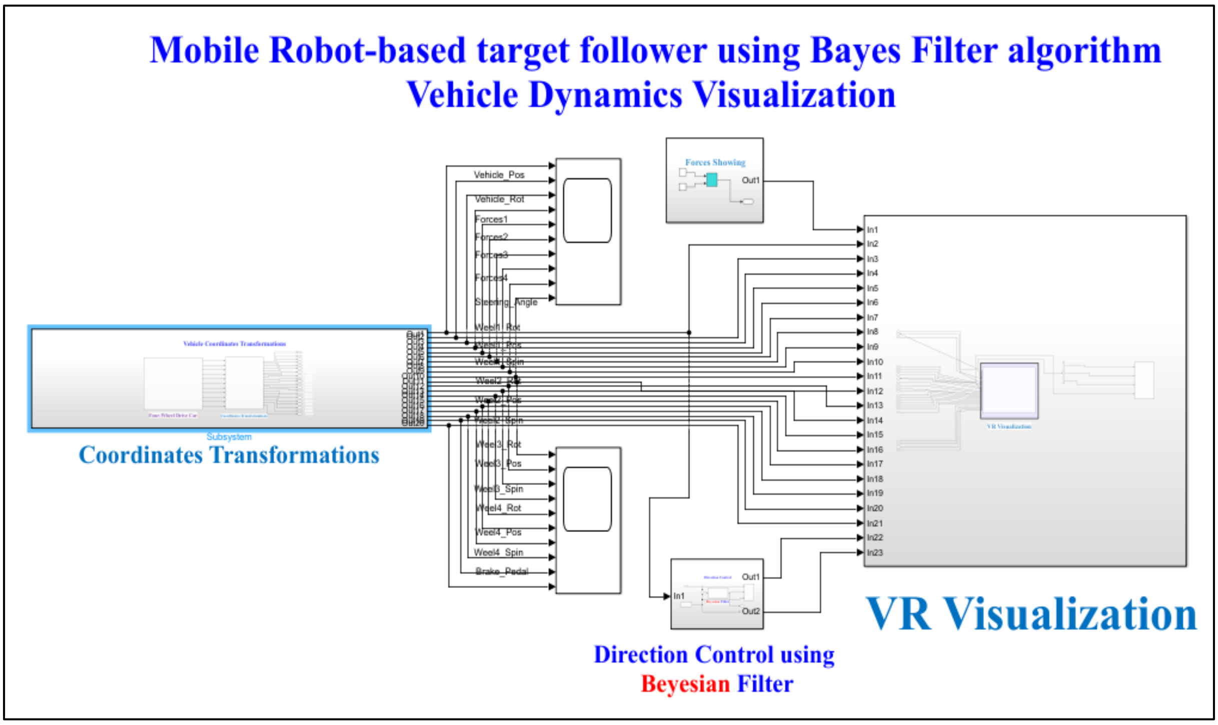
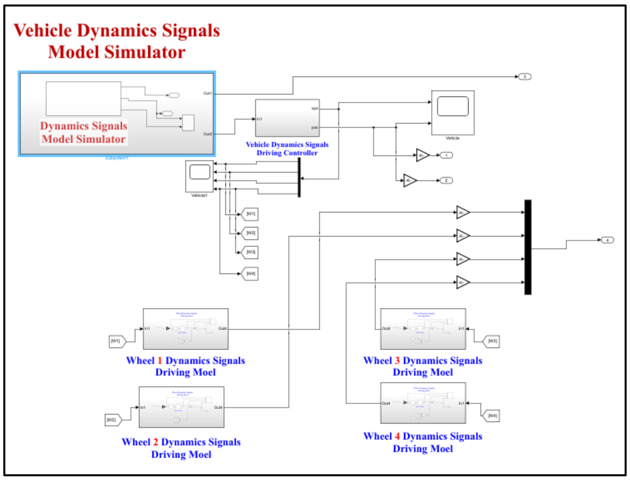
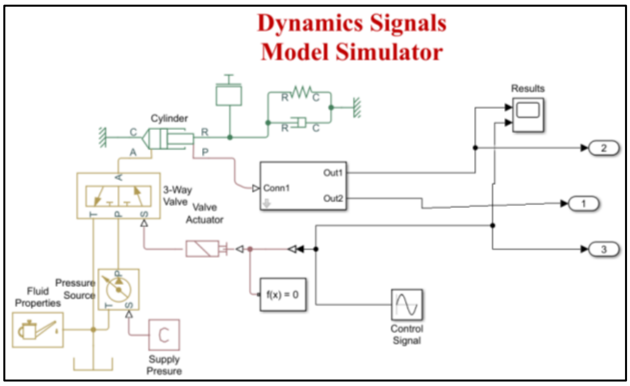


FIGURE 10. **The MATLAB R2018a Simulink block diagram of the robotic vehicle suggested model.**

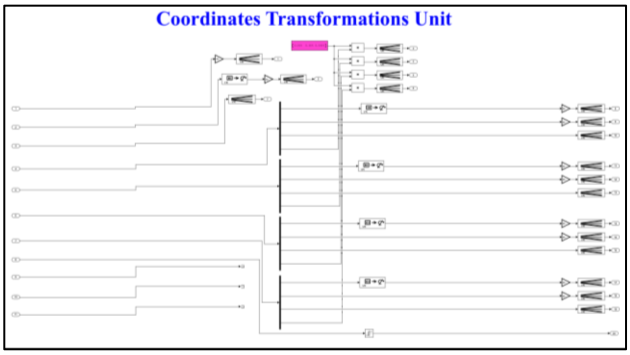
Figure 10 shows the implementation design of the proposed moving vehicle model utilizing the MATLAB application environment. This design includes of the essential blocks which constitute the stages of implementation and operation of the suggested model. We could clarify the details of such units which compose the structure of the recommended scheme for designing the moving vehicle, as shown in Figure 11.


Vehicle Dynamics Signals
Model Simulator
Dynamics Signals
Model Simulator
Vehicle Dynamics Signals
Driving Controller
Wheel 1 Dynamics Signals
Driving Moel
Wheel 3 Dynamics Signals
Driving Moel
Wheel 2 Dynamics Signals
Driving Moel
Wheel 4 Dynamics Signals
Driving Moel

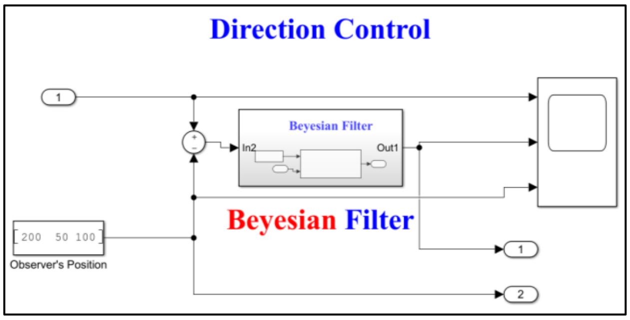
(a)


Dynamics Signals
Model Simulator
Results
Cylinder
3-Way
Valve
Valve
Actuator
Conn1
Out1
Out2
f(x) = 0
Control
Signal
Fluid
Properties
Pressure
Source
Supply
Presure

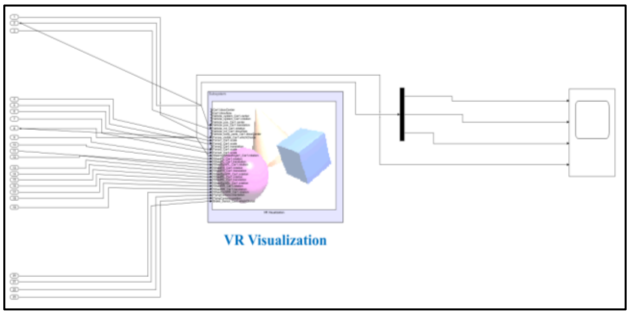
(b)


Coordinates Transformations Unit

(c)


Direction Control
Beyesian Filter
In2
Out1
Beyesian Filter
200 50 100
Observer's Position

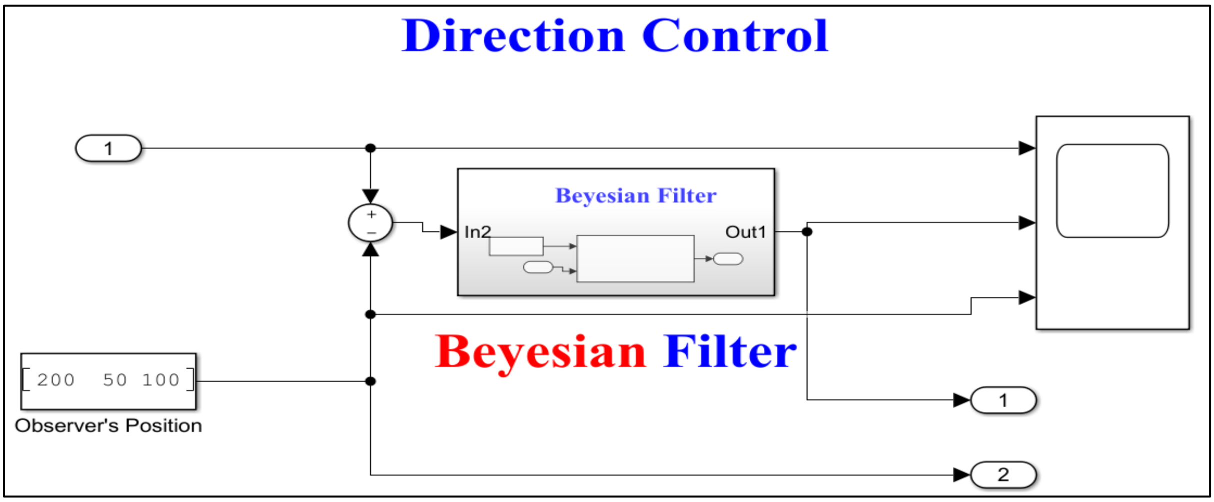
(d)



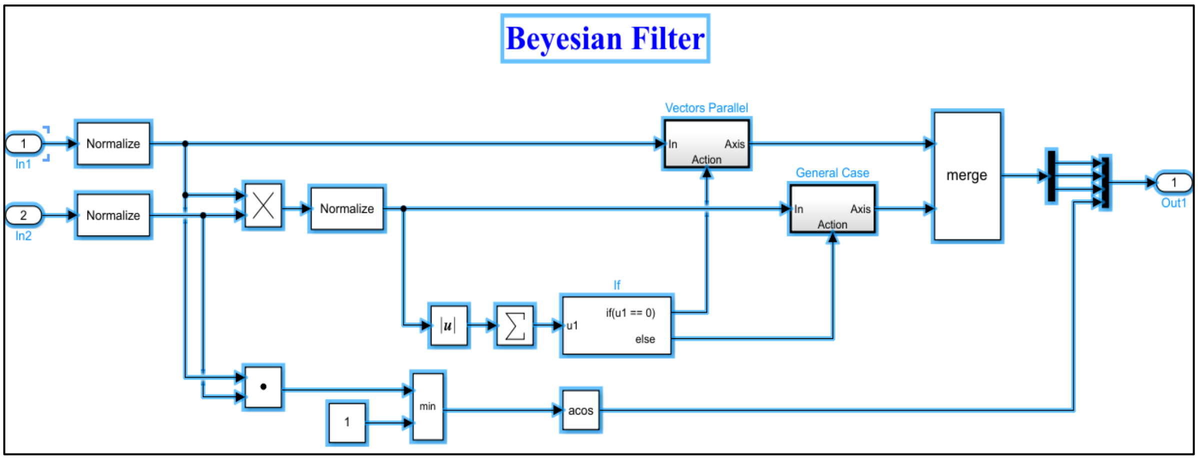
(e)

FIGURE 11. The details of the units composing the structure the robotic vehicle suggested model, (a) Vehicle dynamics simulator, (b) Dynamic signals model simulator, (c) Coordinate Transformation, (d) Beyesian Filter (e) VR realization units.

Thus, regarding the details of Figure 11, which shows the MATLAB simulation of the implementing stages the suggested model for an autonomous robotic vehicle system. We could pay attention to the mechanism for generating and recalling the project's data set, those will be explained in the next paragraphs. The data set includs signals representing the vehicle's motion, which describing the velocity and applied force to the wheels with brakes, as well as vehicle motion control signals which generated from the vehicle dynamics simulator unit introduced in Figure 11 (a). The path map for the track specified for the vehicle's motion is then defined, with the vehicle's specifications, through the dynamic signals simulation unit which shown in Figure 11 (b). This is followed by the stage of measuring and computing the records of the data set signals and analyzing them through coordinate transformation units as displayed in Figure 11 (c). Such units contain blocks for analyzing the transformation matrices of the vehicle movement input signals to prepare them for synchronization with the track map. The coordinate conversion units also contain vision description blocks that work to display the signals describing the vehicle's movement in a way that is close to reality so that they might be observed. The resulting signals are displayed and then the vehicle movement signals are entered into the Bayesian filter unit to estimate and predict the vehicle movement locations more accurately in accordance with the route map information for direction control as shown in Figure 11 (d). At last, all the data resulting along the waves describing the wheel motion and the track map and converted through the previous units of the system are updated and displayed using the virtual reality display unit. This unit consist of a software that analyzes all signals towards a virtual reality video to describe the vehicle's motion track environment, the shape against size of the vehicle, and the expected update motion waves coming from the Bayesian filter with the rest of the transformation matrices and image display units as presented in Figure 11 (e). Next, the Baysian filter scheme in our suggested model, we also utilize the MATLAB R2018a utilities and simulation tools as displayed in Figure 12.



(a)



(b)

FIGURE 12. The Baysian filter scheme in our suggested model, utilizing the MATLAB R2018a utilities and simulation tools, (a) General simulink structure, (b) Detailed simulink structure.

By looking at Figure 12, we could notice the details of the Bayesian filter construction, that includs blocks which representing the probability equations and distinguishing motion with maneuver signals for representing the proposed structure, in addition to distinguishing and predicting visual signals representing the project.

B. The Implemented Dataset

In this paper, the implemented dataset specefied to simulate the mobile robot vehicle have been produced from MATLAB R2018a library and loaded to define all the important mobile robot vehicle dynamic signals.

We could notice that the data set utilized in the research and prepared by the MATLAB environment library represents all the dynamic signals necessary to simulate the movement of the moving vehicle model. These signals represent the vehicle's speed, the vehicle's location, the power supplied to the four wheels, four-wheel rotation signals, signals for settling the positions of the four wheels, and signals for driving these wheels, in addition to braking signals. Such signs were designed according to the four-wheeled moving vehicle simulation model shown in the design aspect of the project and in accordance with the track path designed and prepared by the visual perception unit, which displays the shape of the track for the vehicle's movement with the obstacles, turns and bumps planned.

Furthermore, we could demonstrate a flow chart for the stages of implementing the suggusted model for the autonomous vehicle system, as displayed in Figure 13.

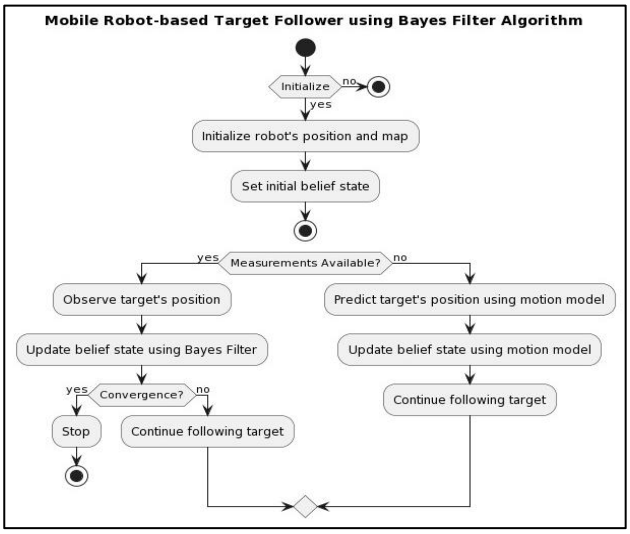


FIGURE 13. The flow chart of implementing the suggested autonomous vehicle system model.

By observing the details of Figure 13, which illustrates the flow chart of the stages of implementing the proposed model for an autonomous vehicle system, we could pay attention to the mechanism for generating and recalling the project's data set, which was previously explained. The data set contains signals describing the vehicle's movement, describing the speed and force applied to the wheels and brakes, and vehicle movement control signals. The path map for the track designated for the vehicle's movement is then defined, as well as the vehicle's specifications, through the four-wheel vehicle simulation system. This is followed by the stage of measuring and calculating the readings of the data set signals and analyzing them utlizing coordinate conversion units. These units contain blocks for analyzing the transformation matrices of the vehicle movement input signals to prepare them for synchronization with the track map. The coordinate conversion units also contain vision description blocks that work to display the signals describing the vehicle's movement in a way that is close to reality so that they can be seen. The resulting signals are displayed and then the vehicle movement signals are entered into the Bayesian filter unit to estimate and predict the vehicle movement locations more accurately in accordance with the route map information. Finally, all the information resulting from the signals describing the wheel movement and the track map and converted from the previous units of the system is updated and displayed using the virtual reality display unit. This unit consists of software that analyzes all signals into a virtual reality to describe the vehicle's movement track environment, describe the shape and size of the vehicle, and update its expected movement signals coming from the Bayesian filter and the rest of the transformation matrices and image display units.

IV. Results & Discussion

By implementing the proposed model of robotic vehicle based on Beyesian filter, we might follow the implementation of the program and present the resulting signals depending to sequence that have been configured. First of all, the vehicle speeds and position signals obtained along the vehicle dynamics signals and control model simulator are illustrated in Figure 14.

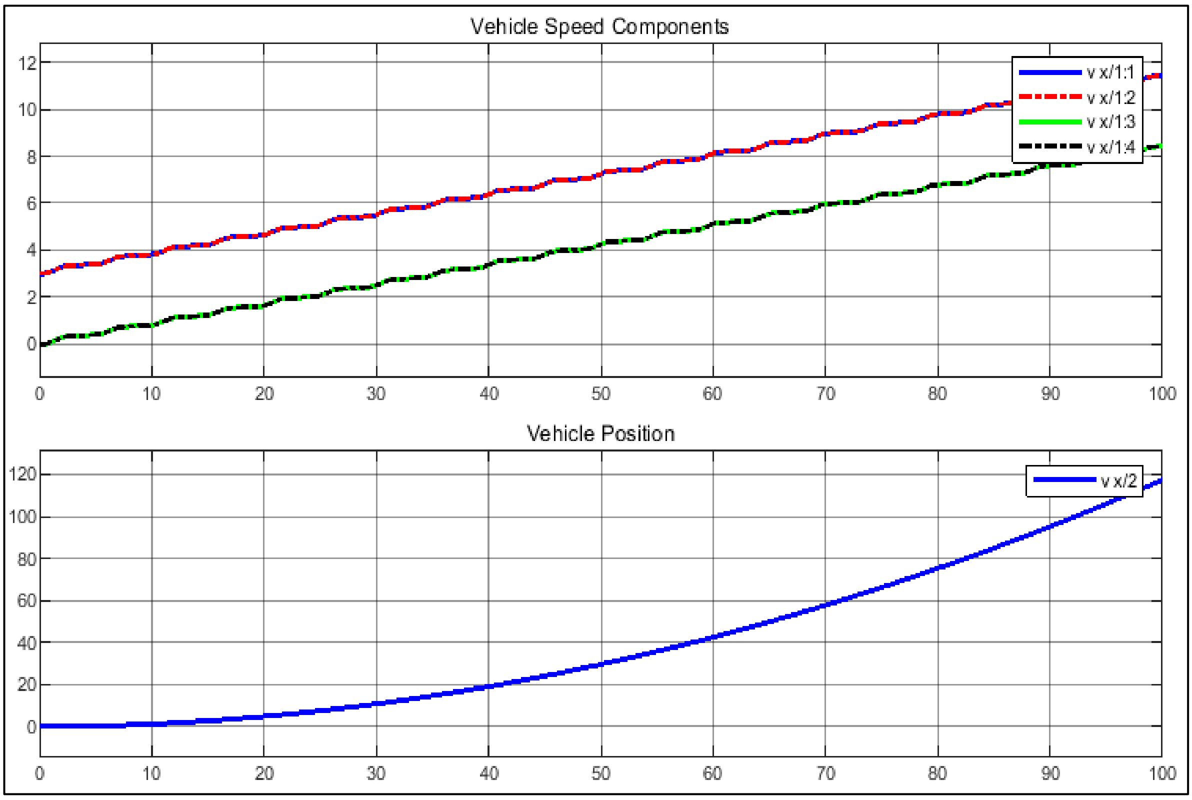
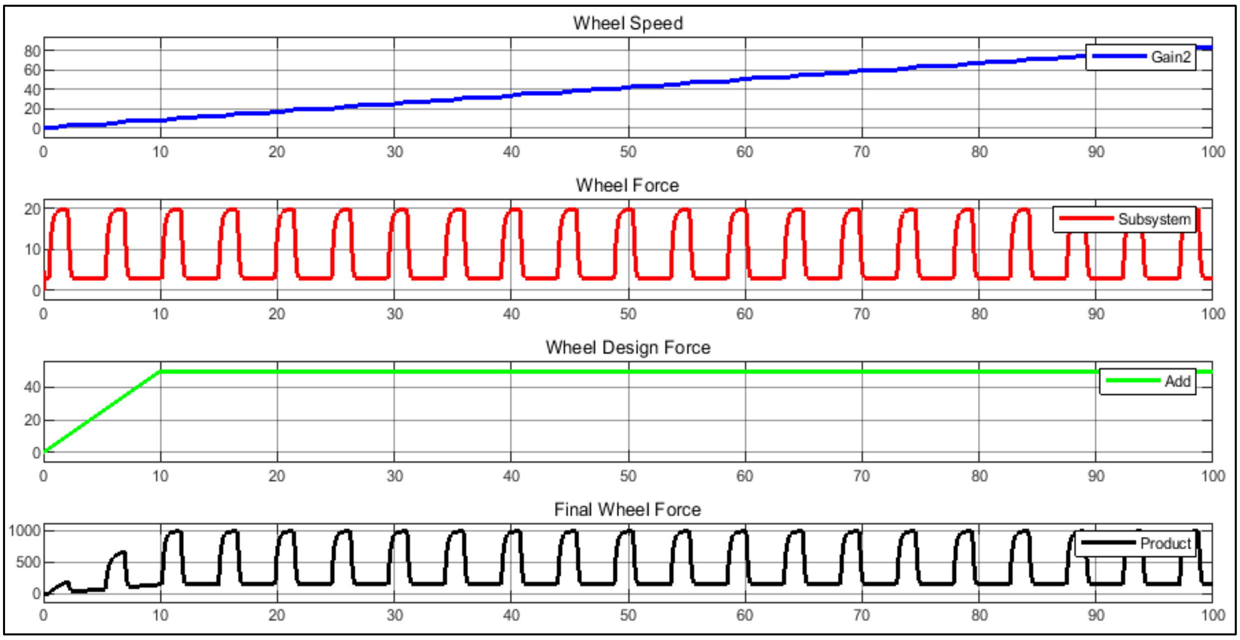
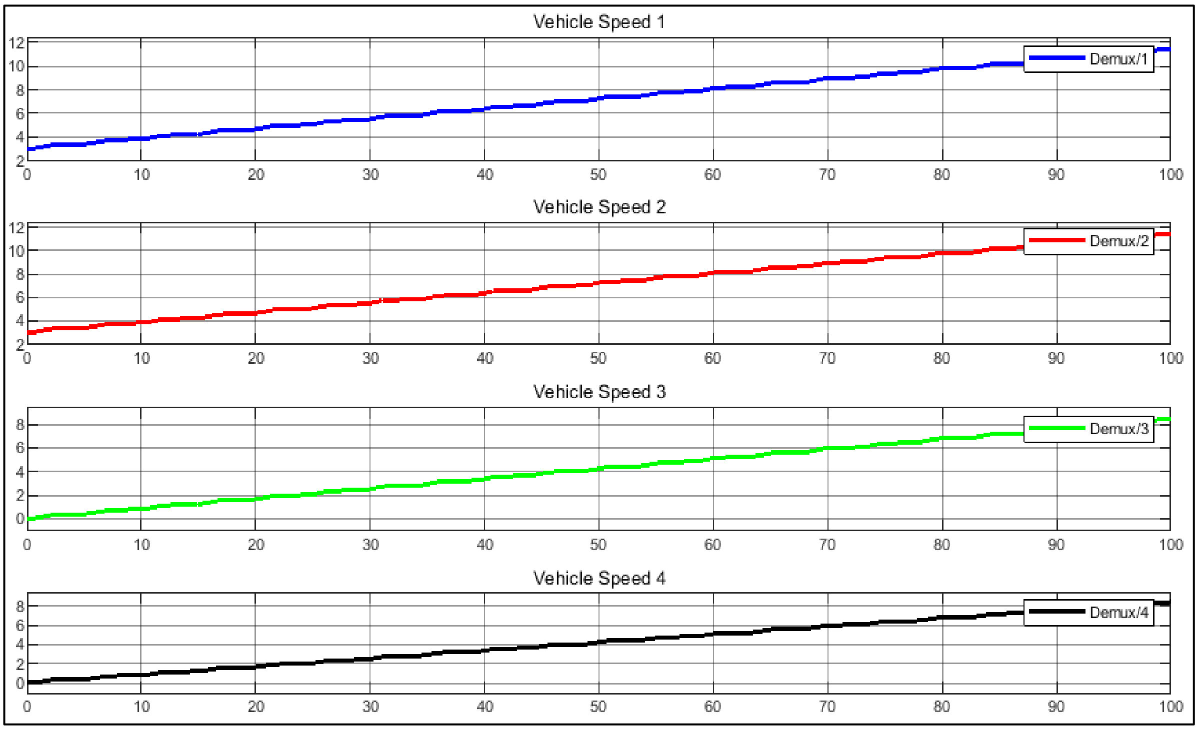


FIGURE 14. **The vehicle velocities and position signals achieved from the vehicle dynamics signals and control model simulator.**

By recognizing the signals of Figure 14, that indicate the vehicle's wheel velocity signals in addition to the vehicle's position signal. The source of such signals obtained from the vehicle's dynamic motions simulation unit. The velocity signals of the simulated vehicle's wheels varied progressively against time until they reach the permissible values, while the vehicle's location signals varied non-linearly with time in accordance with the project's design path map. Next, the simulated robotic vehicle velocity signals of the four wheels are demonstrated in Figure 15.



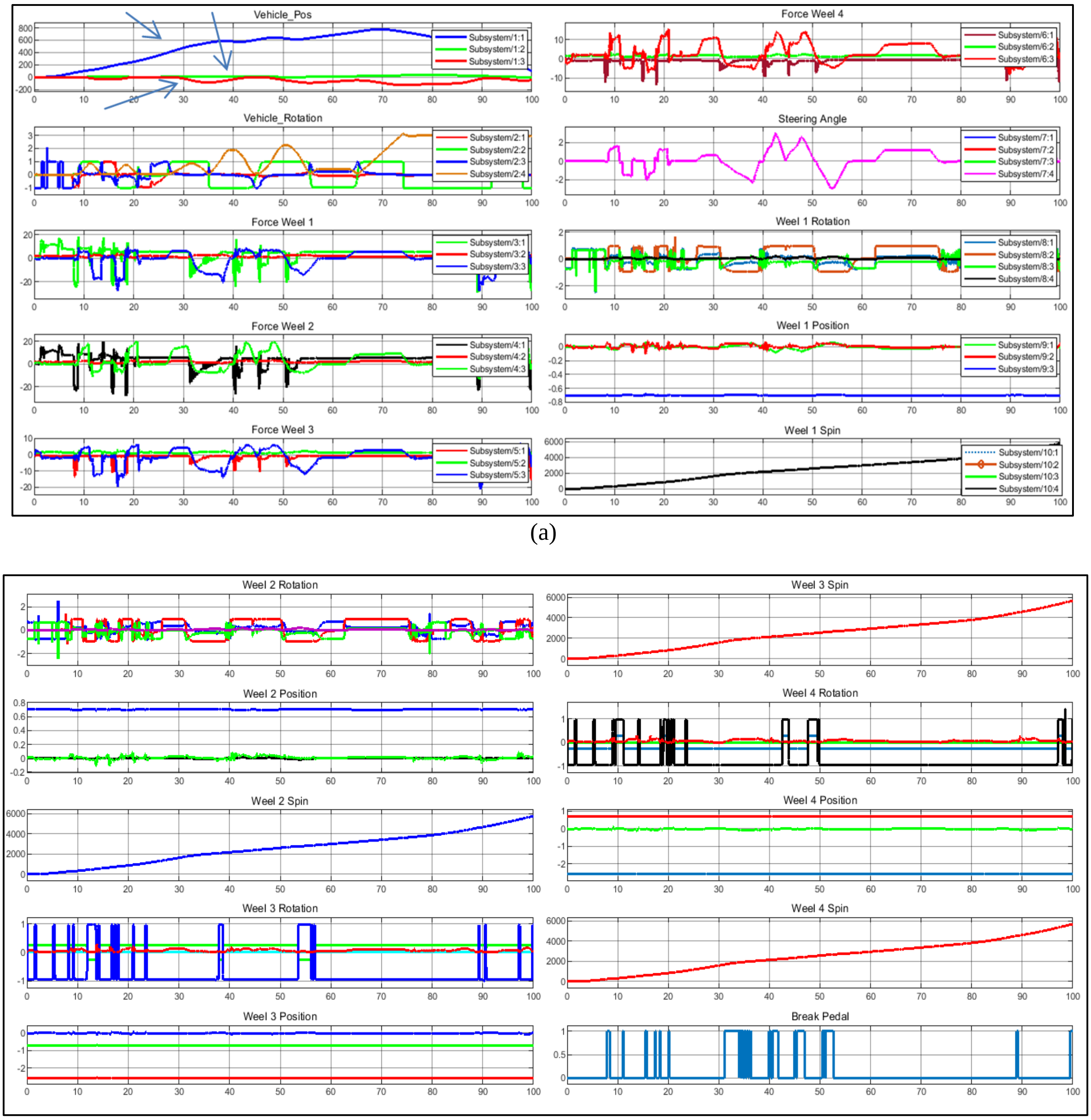
(a)



(b)

FIGURE 15. The simulated robotic vehicle speed signals of the four wheels, (a) Wheel dynamics signals driving model signals, (b) Vehicle dynamic 4-wheels signals.

Also, by looking at the results of Figure 15, we might observe the mechanism of action of the vehicle wheel simulation mass signals, such as force and torque signals for each wheel, through which the speed signal for each wheel is extracted as shown in Figure 15 (a). We could also observe the results of the speed signals resulting from The mass simulates the operation of the vehicle's wheel, which increases linearly until the maximum limit is reached for all wheels, as shown in Figure 15 (b). Actually, these signals will be loaded, in accordance with the rest of the mobile vehicle motions simulation signals arranged by the data set, toward the coordinate converter simulation unit, those will analyze them and set their coordinates to be compatible and synchronized along the coordinates of the motion path map prepared by the data set for this research. Now, the obtained vehicle dynamic signals along the coordinates transformation unit have been evaluated as displayed in Figure 16.



(a)

(b)

FIGURE 16. **Vehicle coordinate transformation navigation signals, (a) Vehicle position and location with first wheel forces signals, (b) Remaining Wheels forces with brake pedal signals.**

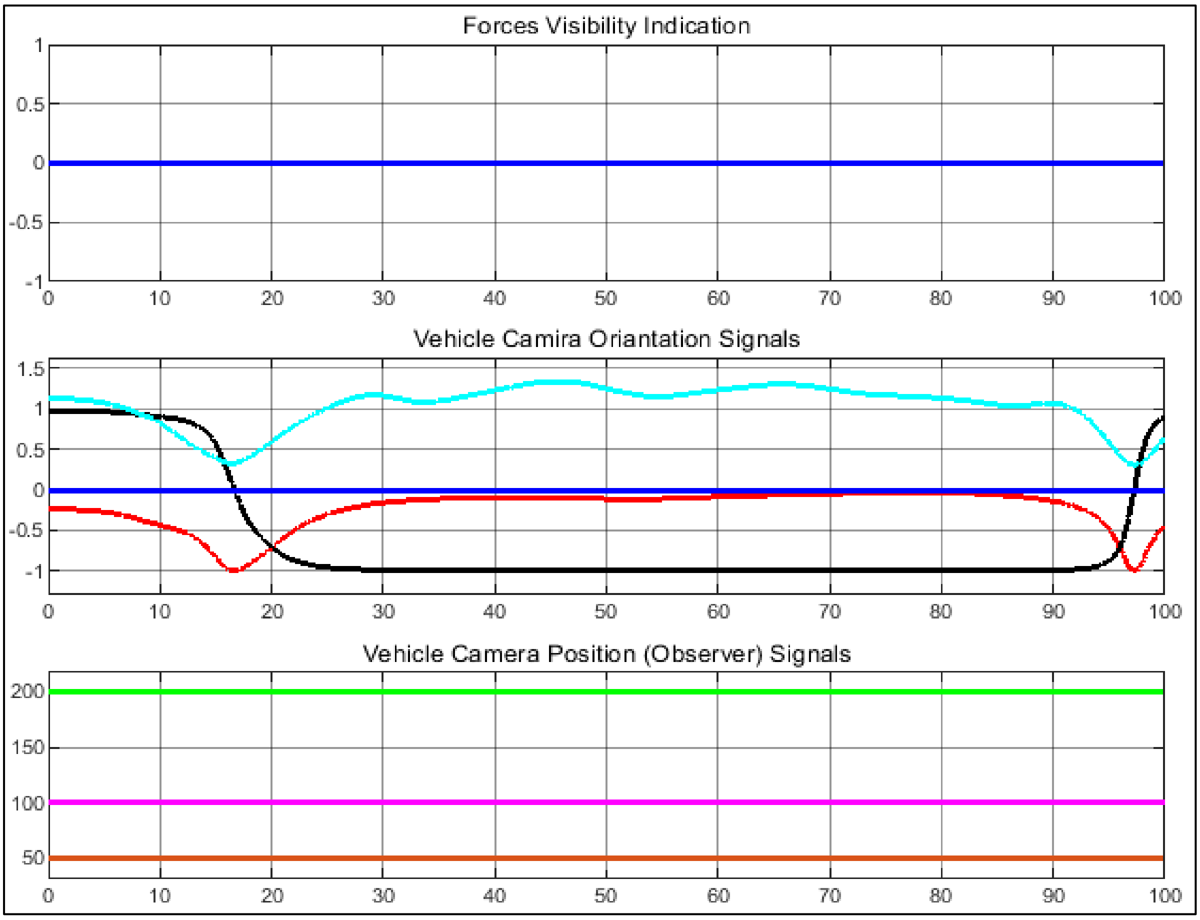
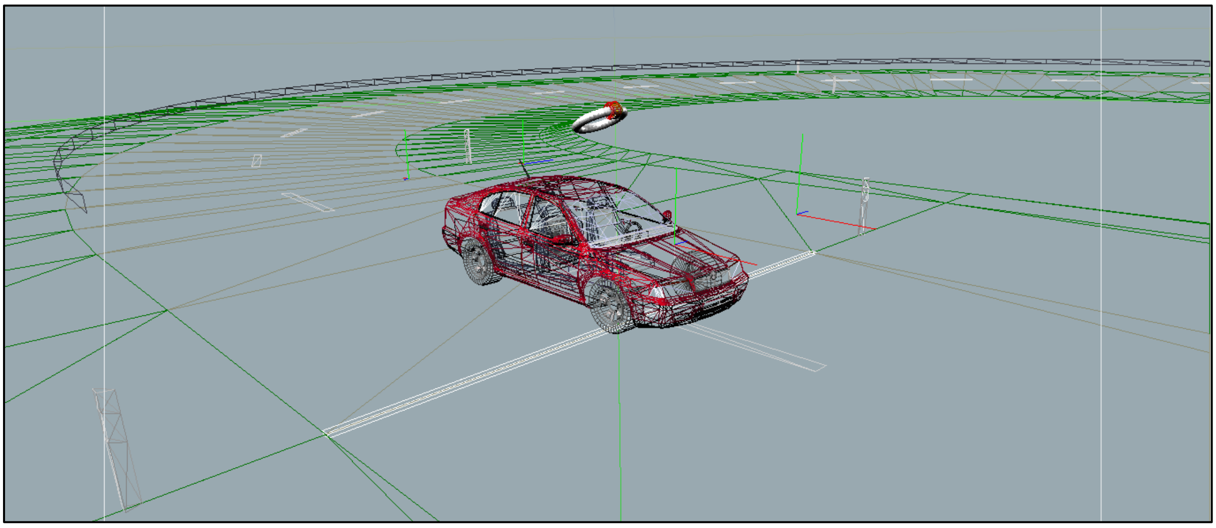
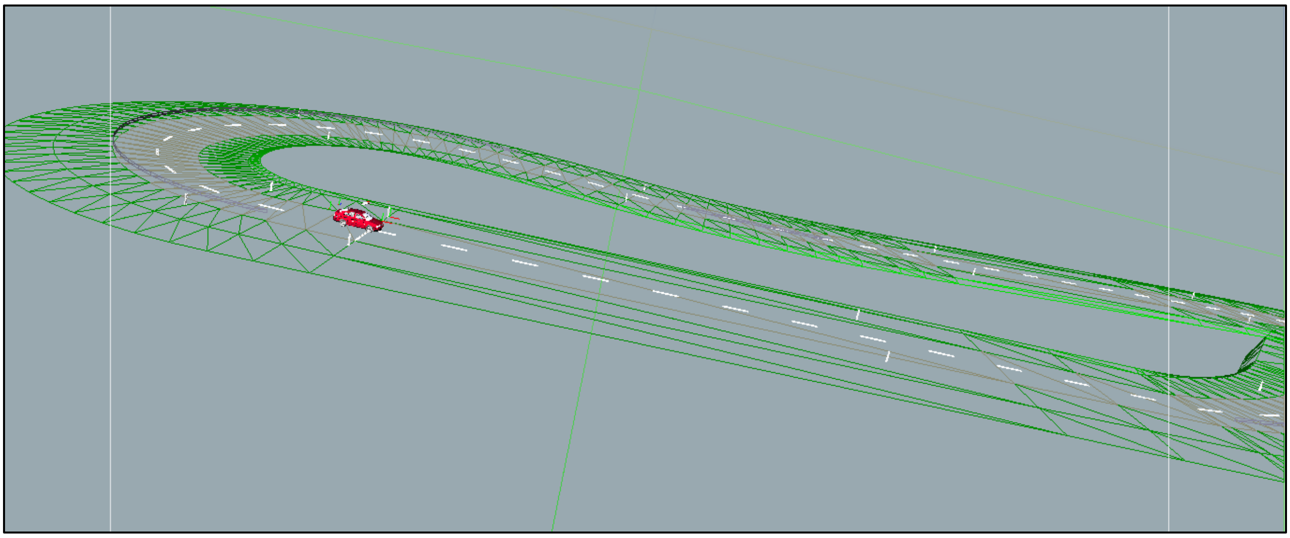


FIGURE 17. **Figure 17: The obtained controlled signals of Beyesian filter.**

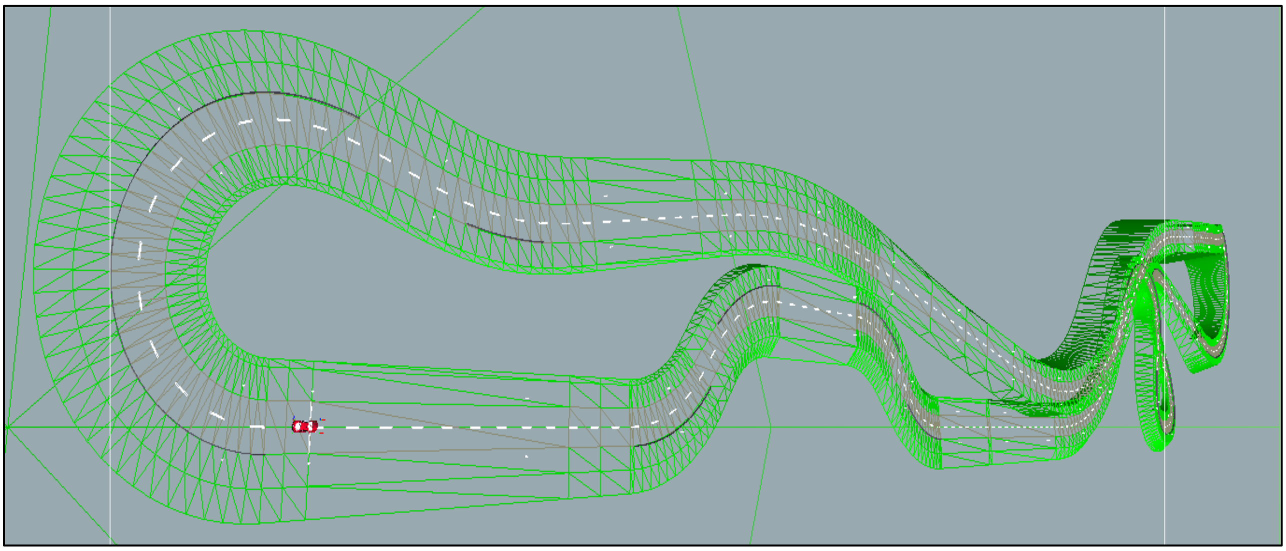
Thus, by regarding Figure 17, we might observe the outcomes of the Bayesian filter results which show the expected amounts of the vehicle's location signals and the pulses of the viewing cameras' locations for the virtual track environment after accessing along the analysis and statistics algorithms and probability equations simulation functions relying on which the Bayesian filter operates to totally control them and advance the implementation accuracy. Furthermore, the signal outcomes are simulated, analyzed, and demonstrated along the virtual reality software display unit, as configured in Figure 18.



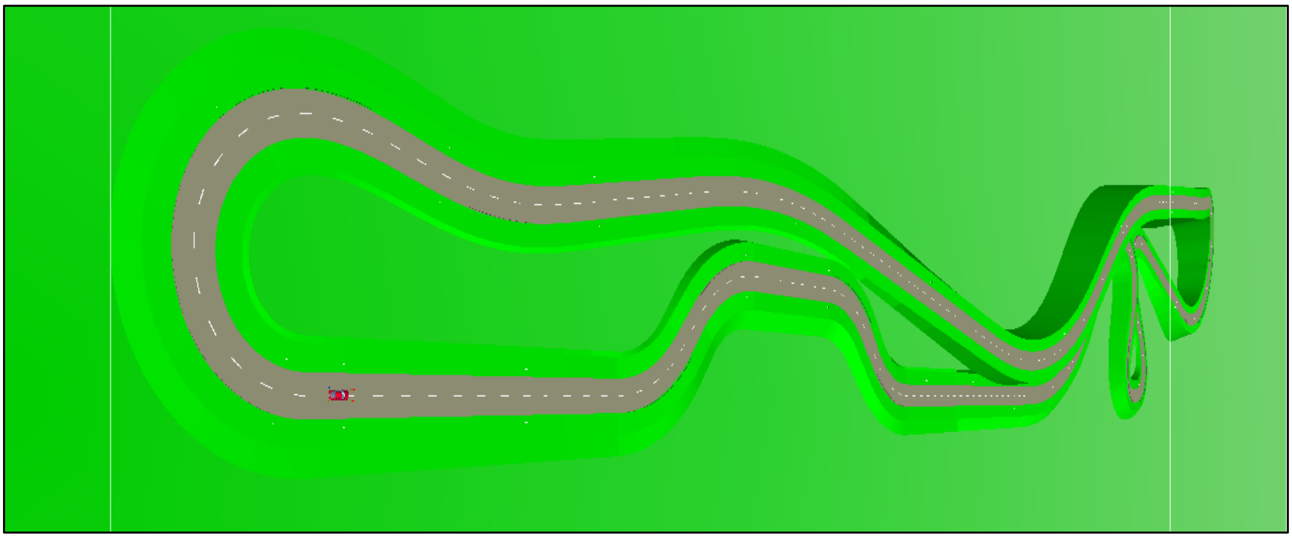
(a)



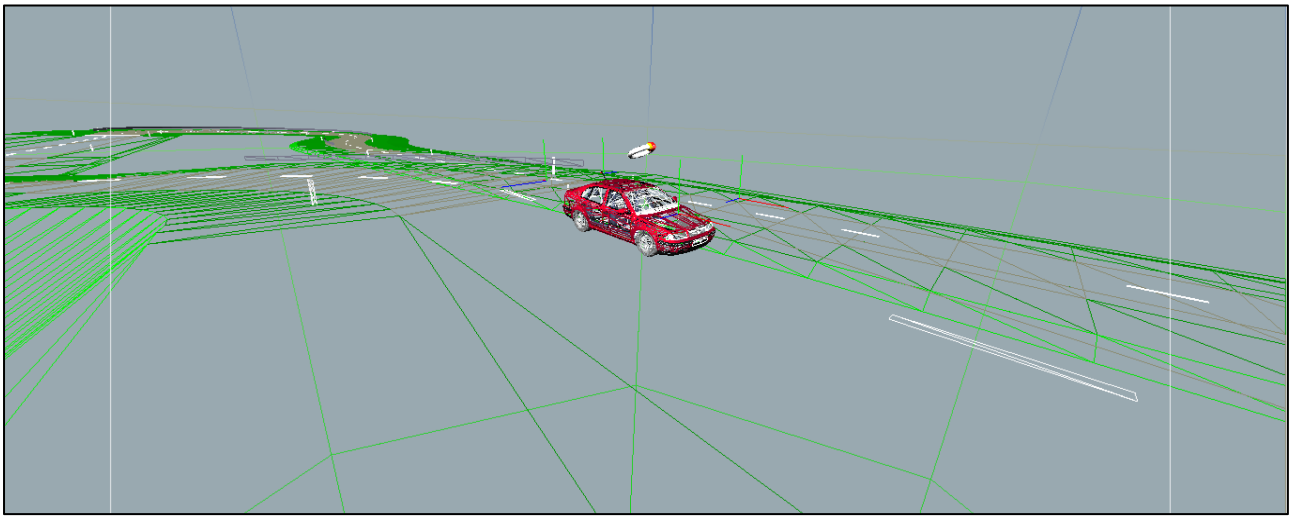
(b)



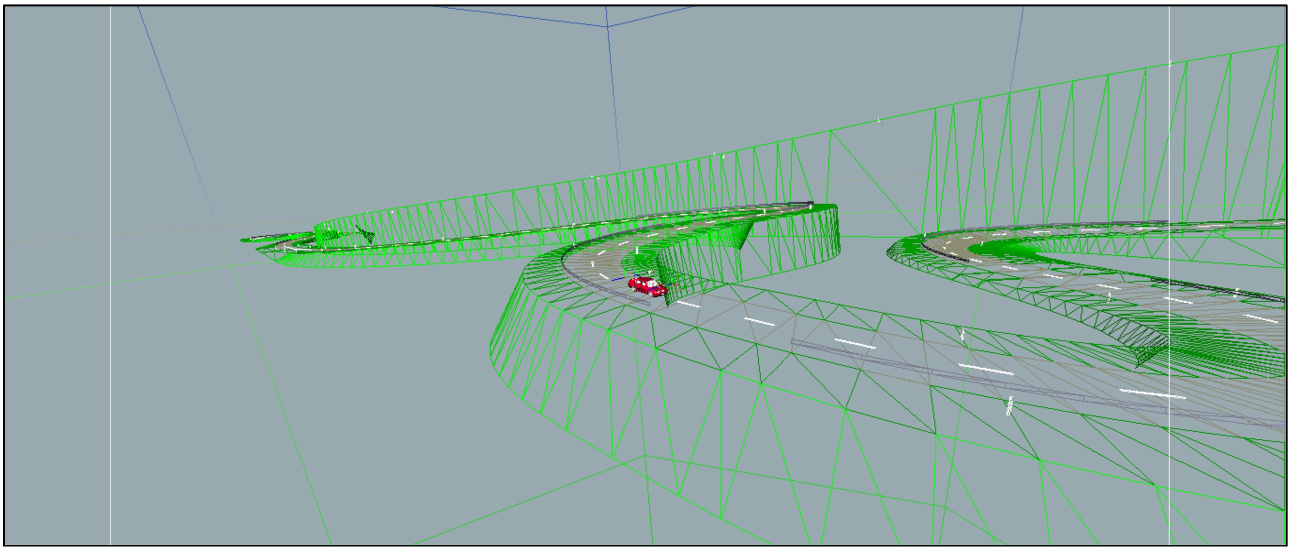
(c)



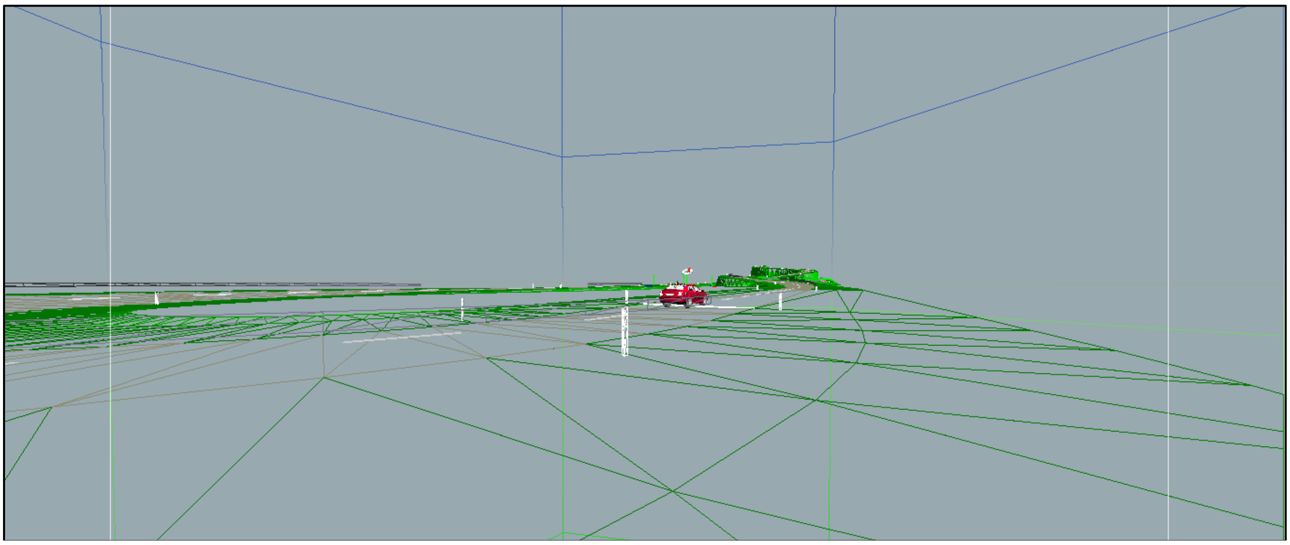
(d)



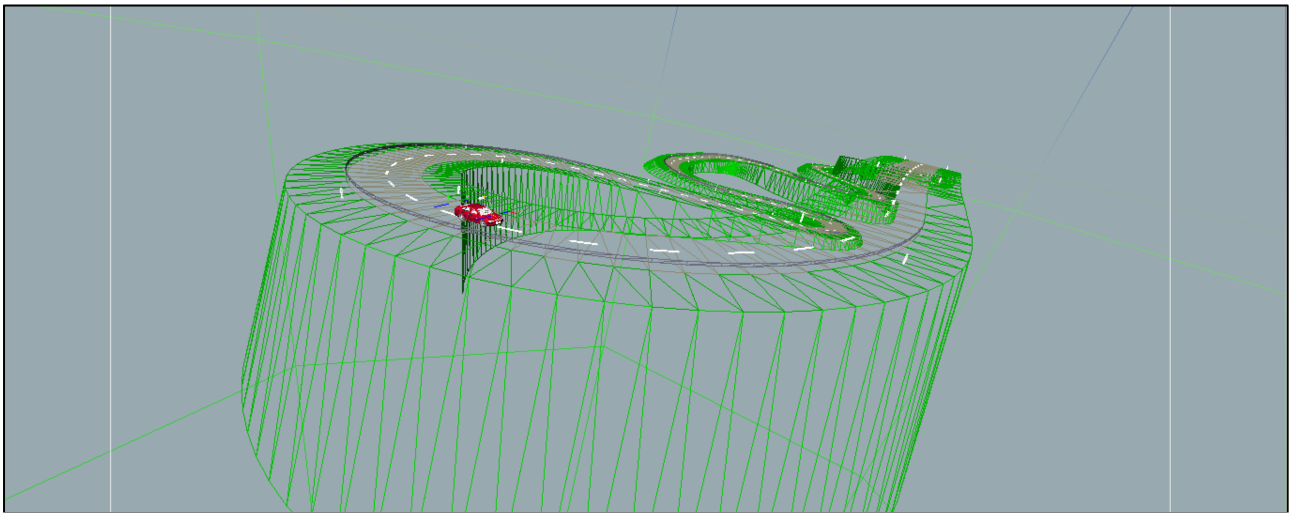
(e)



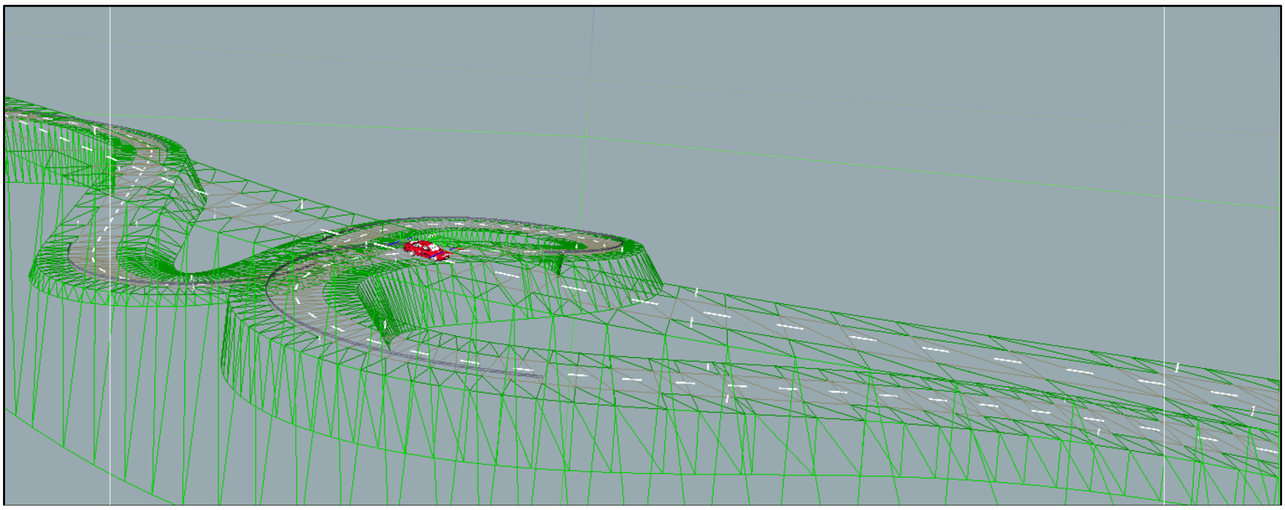
(f)



(g)



(h)



(i)

(j)

FIGURE 18. **Vehicle dynamic visualization navigation wizard inside view, (a) Front view, (b) Far front view, (c) The xy mesh view, (d) Front xy solid view, (e) Front mesh view, (f) Front 3D mesh view, (g) Back 3D mesh view, (h) Isometric mesh view1, (i) Isometric mesh view2, (j) Inside vehicle view.**

By looking at Figure 18, that displays the results of the navigation wizard for the dynamic visualization of the vehicle interior. This application displays, analyzes and simulates the control signals results along a virtual reality software display unit. Figure 18 (a) illustrates the resuls of the mapping signals in front view, while Figure 18 (b) presents the mapping signals outcomes in far front view. Also, Figure 18 (c) outlies the mapped controlled signals outcomes in form of mesh xy view, wheras Figure 18 (d) displays the obtained mapped signals ourcomes in solid front xy view. Moreover, Figure 18 (e) displays the controlling mapped signals results in front mesh view, on the other hand, Figure 18 (f) illustrates the same mapping control signals in front 3D mesh view. Finally, Figure 18 (g) outlines the navigation mapped signals in back 3D mesh view, with isometric mesh view 1 as presented in Figure 18 (h), and isometric mesh view 2 as introduced in Figure 18 (i). At last, Figure 18 (j) shows the navigation control signals from interior vehicle view point. Moreover, the resulting error between the predicted and measured Simulated vehicle position, might be found utilizing the mean square erroe relation which has been computed as shown in Table 1.

TABLE I
THE ACHIEVED MSE AMIDST THE PREDICTED AND MEASURED SIMULATED VEHICLE POSITION AMOUNTS.

Vehicle Position	Simulation Readings (meters)	MSE	
Measured	120	-	-
Predicted	120.0215	0.0215	FP
	119.962	0.038	FN

In this research, the simulation and implementation results of the suggested model on simulating the path environment of an autonomous vehicle with the assistance of a Bayesian filter are reviewed and explained. Implementation results of the proposed Bayesian filter-based autonomous mobile car robot model were extracted and reviewed using the MATLAB application environment. The performance of the proposed model was implemented through the capabilities of the MATLAB application environment, which provides efficient software functions for simulating the details of the mobile robot and vehicle with the path planner, also producing imaging capabilities through different directions and apparatus for analyzing and displaying signals with the assistance of the MATLAB library features. In this paper, the signal flow operations for dynamic simulation are also clarified, explained, interpreted and analyzed to illustrate their motion and impacts. The achieved results were also compared and verified by examining their efficiency and accuracy level of implementation. Furthermore, the implemented proposed model quality verification measures have been computed and listed in Table 2.

TABLE II
THE QUALITY VERIFICATION MEASURES OF IMPLEMENTED PROPOSED MODEL.

Simulation Quality Measurements		Accuracy %	Specificity %	Sensitivity %	Precision %	F score %
TP	0.9785	97.025	97.814	96.2616	97.850	97.0493
TN	0.962					
FP	0.0215					
FN	0.038					

Comparing our results with those from other research shows that our study's outcomes are very strong. For instance, paper [51] reports an error rate of 0.5% to 10% across various parts of the path. Another study [52] found that for person images, YOLO achieves a total detection rate of 93%, while other objects have a detection rate between 90% and 92%. With the SSD model, person sample frames from a top view have a true positive rate of 93%, and other objects reach up to 90%. Using different tracking algorithms for multiple object tracking, the tracking accuracy with both detection models ranges from 90% to 94%.

V. Conclusion

This research presented a new approach to object tracking, leveraging the Bayes filter algorithm within the constraints of camera setup and viewing angles. The integration of the Bayes filter algorithm provides a probabilistic framework, effectively addressing the challenges posed by occlusions, illumination changes, and unexpected object movements in real-world scenarios. In this article, the simulation and implementation results of the proposed mobile vehicle robot model based on Bayesian filter are investigated and analyzed. The performance of the proposed model was implemented through the capabilities of the MATLAB R2018a application environment, with a dynamic simulation efficiency of 97.025%, sensitivity of 96.2616%, and overall system quality (F-score) of 97.0493%.

REFERENCES

[1] Chowdhury, A., et al. (2020). "Attacks on self-driving cars and their countermeasures: A survey." IEEE Access 8: 207308-207342.

[2] Ni, J., et al. (2020). "A survey on theories and applications for self-driving cars based on deep learning methods." Applied Sciences 10(8): 2749.

[3] Stilgoe, J. (2018). "Machine learning, social learning and the governance of self-driving cars." Social studies of science 48(1): 25-56.

[4] Soleimanitaleb, Z. and M. A. Keyvanrad (2022). "Single object tracking: A survey of methods, datasets, and evaluation metrics." arXiv preprint arXiv:2201.13066.

[5] Luo, W., et al. (2021). "Multiple object tracking: A literature review." Artificial intelligence 293: 103448.

[6] Xu, Y., et al. (2019). "Deep learning for multiple object tracking: a survey." IET Computer Vision 13(4): 355-368.

[7] Yuan, D., et al. (2020). "Self-supervised deep correlation tracking." IEEE Transactions on Image Processing 30: 976-985.

[8] Kowalek, P., et al. (2019). "Classification of diffusion modes in single-particle tracking data: Feature-based versus deep-learning approach." Physical Review E 100(3): 032410.

[9] Tu, Z., et al. (2019). "A survey of variational and CNN-based optical flow techniques." Signal Processing: Image Communication 72: 9-24.

[12] Vagale, A.; Oucheikh, R.; Bye, R.T.; Osen, O.L.; Fossen, T.I. Path planning and collision avoidance for autonomous surface vehicles I: A review. J. Mar. Sci. Technol. 2021, 26, 1292–1306.

[13] Souissi, O.; Benatitallah, R.; Duvivier, D.; Artiba, A.; Belanger, N.; Feyzeau, P. Path planning: A 2013 survey. In Proceedings of the 2013 International Conference on Industrial Engineering and Systems Management (IESM), Rabat, Morocco, 28–30 October 2013; pp. 1–8.

[14] Yi, C.; Jeong, S.; Cho, J. Map representation for robots. Smart Comput. Rev. 2012, 2, 18–27.

[15] Nash, A.; Koenig, S. Any-angle path planning. AI Mag. 2013, 34, 85–107.

[16] Petres, C.; Pailhas, Y.; Petillot, Y.; Lane, D. Underwater path planing using fast marching algorithms. In Proceedings of the Oceans 2005-Europe, Brest, France, 20–23 June 2005; Volume 2, pp. 814–819.

[17] Huang, H.P.; Chung, S.Y. Dynamic visibility graph for path planning. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2813–2818.

[18] Bergman, K.; Ljungqvist, O.; Axehill, D. Improved path planning by tightly combining lattice-based path planning and optimal control. IEEE Trans. Intell. Veh. 2020, 6, 57–66.

[19] Papadakis, P. Terrain traversability analysis methods for unmanned ground vehicles: A survey. Eng. Appl. Artif. Intell. 2013, 26, 1373–1385.

[20] Zhang, H.; Zhang, Y.; Yang, T. A survey of energy-efficient motion planning for wheeled mobile robots. Ind. Robot. Int. J. Robot. Res. Appl. 2020, 47, 607–621.

[21] Effati, M.; Fiset, J.S.; Skonieczny, K. Considering slip-track for energy-efficient paths of skid-steer rovers. J. Intell. Robot. Syst. 2020, 100, 335–348.

[22] Patel, N.; Slade, R.; Clemmet, J. The ExoMars rover locomotion subsystem. J. Terramech. 2010, 47, 227–242.

[23] Brunner, M.; Fiolka, T.; Schulz, D.; Schlick, C.M. Design and comparative evaluation of an iterative contact point estimation method for static stability estimation of mobile actively reconfigurable robots. Robot. Auton. Syst. 2015, 63, 89–107.

[24] Sánchez-Ibánez, J.R.; Pérez-del Pulgar, C.J.; Azkarate, M.; Gerdes, L.; García-Cerezo, A. Dynamic path planning for reconfigurable rovers using a multi-layered grid. Eng. Appl. Artif. Intell. 2019, 86, 32–42.

[25] Norouzi, M.; Miro, J.V.; Dissanayake, G. Planning stable and efficient paths for reconfigurable robots on uneven terrain. J. Intell. Robot. Syst. 2017, 87, 291–312.
[26] Barjuei, E.S.; Boscariol, P.; Vidoni, R.; Gasparetto, A., "Robust control of three-dimensional compliant mechanisms". J. Dyn. Syst. Meas. Control 2016, 138, 101009. [CrossRef].
[27] Rsetam, K.; Cao, Z.; Man, Z. Hierarchical non-singular terminal sliding mode controller for a single link flexible joint robot manipulator. In Proceedings of the IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, VIC, Australia, 12–15 December 2017; pp. 6677–6682.
[28] Shokoohinia, M.R.; Fateh, M.M. Robust dynamic sliding mode control of robot manipulators using the Fourier series expansion. Trans. Inst. Meas. Control 2019, 41, 2488–2495. [CrossRef]
[29] Soltanpour M. R., Khooban M. H. A particle swarm optimization approach for fuzzy sliding mode control for tracking the robot manipulator. Nonlinear Dynamics, Vol. 74, Issue 1, 2013, p. 467-478.
[30] Mahmoodabadi, M.; Nejadkourki, N. Trajectory Tracking of a Flexible Robot Manipulator by a New Optimized Fuzzy Adaptive Sliding Mode-Based Feedback Linearization Controller. J. Robot. 2020, 2020, 8813217. [CrossRef]
[31] Zaare, S.; Soltanpour, M.R.; Moattari, M. Adaptive sliding mode control of n flexible-joint robot manipulators in the presence of structured and unstructured uncertainties. Multibody Syst. Dyn. 2019, 47, 397–434. [CrossRef]
[32] Tuan, H.M.; Sanfilippo, F.; Hao, N.V. Modelling and Control of a 2-DOF Robot Arm with Elastic Joints for Safe Human-Robot Interaction. Front. Robot. AI 2021, 8, 679304. [CrossRef]
[33] Tuan, H.M.; Sanfilippo, F.; Hao, N.V. An Adaptive Sliding Mode Controller for a 2-DOF Elastic Robotic Arm. In Proceedings of the 4th International Conference on Intelligent Technologies and Applications (INTAP 2021), Grimstad, Norway, 11–13 October 2021.
[34] He, Q., et al. (2022). "Multi-object tracking in satellite videos with graph-based multitask modeling." IEEE Transactions on Geoscience and Remote Sensing 60: 1-13.
[35] Zhu, P., et al. (2021). "Detection and tracking meet drones challenge." IEEE Transactions on Pattern Analysis and Machine Intelligence 44(11): 7380-7399.
[36] Raghavan, V.S.; Kanoulas, D.; Laurenzi, A.; Caldwell, D.G.; Tsagarakis, N.G. Variable configuration planner for legged-rolling obstacle negotiation locomotion: Application on the centauro robot. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 4738–4745.
[37] Otsu, K.; Matheron, G.; Ghosh, S.; Toupet, O.; Ono, M. Fast approximate clearance evaluation for rovers with articulated suspension systems. J. Field Robot. 2020, 37, 768–785.
[38] Hines, T.; Stepanas, K.; Talbot, F.; Sa, I.; Lewis, J.; Hernandez, E.; Kottege, N.; Hudson, N. Virtual Surfaces and Attitude Aware Planning and Behaviours for Negative Obstacle Navigation. IEEE Robot. Autom. Lett. 2021, 6, 4048–4055.
[39] Taghavifar, H.; Rakheja, S.; Reina, G. A novel optimal path-planning and following algorithm for wheeled robots on deformable terrains. J. Terramech. 2020, 96, 147–157.
[40] Arvidson, R.E.; Bell, J.; Bellutta, P.; Cabrol, N.A.; Catalano, J.; Cohen, J.; Crumpler, L.S.; Des Marais, D.; Estlin, T.; Farrand,W.; et al. Spirit Mars Rover Mission: Overview and selected results from the northern Home Plate Winter Haven to the side of Scamander crater. J. Geophys. Res. Planets 2010, 115, 1–19. [CrossRef]
[41] Ishigami, G.; Nagatani, K.; Yoshida, K. Path planning for planetary exploration rovers and its evaluation based on wheel slip dynamics. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 2361–2366.
[42] Sutoh, M.; Otsuki, M.; Wakabayashi, S.; Hoshino, T.; Hashimoto, T. The right path: Comprehensive path planning for lunar exploration rovers. IEEE Robot. Autom. Mag. 2015, 22, 22–33.
[43] Inotsume, H.; Creager, C.; Wettergreen, D.; Whittaker, W. Finding routes for efficient and successful slope ascent for exploration rovers. In Proceedings of the 13th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Beijing, China, 19–22 June 2016; pp. 1–10.
[44] Inotsume, H.; Kubota, T.; Wettergreen, D. Robust Path Planning for Slope Traversing under Uncertainty in Slip Prediction. IEEE Robot. Autom. Lett. 2020, 5, 3390–3397.

International Journal of Multiphysics
Volume 18, No. 3, 2024
ISSN: 1750-9548

[45] Niksirat, P.; Daca, A.; Skonieczny, K. The effects of reduced-gravity on planetary rover mobility. Int. J. Robot. Res. 2020, 39, 797–811.

[46] Plonski, P.A.; Tokekar, P.; Isler, V. Energy-efficient path planning for solar-powered mobile robots. J. Field Robot. 2013, 30, 583–601.

[47] Kaplan, A.; Kingry, N.; Uhing, P.; Dai, R. Time-optimal path planning with power schedules for a solar-powered ground robot. IEEE Trans. Autom. Sci. Eng. 2016, 14, 1235–1244.

[48] Groves, K.; Hernandez, E.; West, A.; Wright, T.; Lennox, B. Robotic Exploration of an Unknown Nuclear Environment Using Radiation Informed Autonomous Navigation. Robotics 2021, 10, 78.

[49] Farag, W. (2021). "Kalman-filter-based sensor fusion applied to road-objects detection and tracking for autonomous vehicles." Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 235(7): 1125-1138.

[50] Daníş, F. S., et al. (2021). "Adaptive sequential Monte Carlo filter for indoor positioning and tracking with bluetooth low energy beacons." IEEE Access 9: 37022-37038.

[51] SHAH, H., et al., Vision Based Obstacle Avoidance for Mobile Robot using Optical Flow Process. international Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019. 8(4S): p. 466-470.

[52] Hu, Y., et al., Object detection algorithm for wheeled mobile robot based on an improved yolov4. Applied Sciences, 2022. 12(9): p. 4769.

[53] Rehman, A., Shah, S. A. H., Nizamani, A. U., Ahsan, M., Baig, A. M., & Sadaqat, A. (2024). AI-Driven Predictive Maintenance for Energy Storage Systems: Enhancing Reliability and Lifespan. PowerTech Journal, 48(3). https://doi.org/10.XXXX/powertech.v48.113:contentReference[oaicite:0]{index=0}