Design and Development of Terrain Globetrotter Bot for Different types of Engg. Applications

[1]A. Joseph Walter, [1]Akshay D. Akamanchi, [1]C. Karthik, [1]Mangala Shashank,
[2]Dr. Pavithra G., [3]Dr. T.C.Manjunath* **Ph.D. (IIT Bombay)**

[1]Eighth Semester BE (ECE) Students, Final Year, Dept. of Electronics & Communication Engg.,
Dayananda Sagar College of Engineering, Bangalore, Karnataka
[2]Associate Prof., Electronics & Communication Engg. Dept.,
Dayananda Sagar College of Engineering, Bangalore, Karnataka
[3] Professor & HOD, Electronics & Communication Engg. Dept.,
Dayananda Sagar College of Engineering, Bangalore, Karnataka

*Corresponding author : tcmanju@iitbombay.org

Abstract

The work done in this paper gives the design and development of terrain globetrotter bot (robot) for different types of engg. applications. we are going to design and develop a Terrain Globetrotter Bot. A Terrain Globetrotter is a simple structure bot that uses the ROS (Robotic Operating System) software library of version ROS Kinetic/ Melodic and is booted with a Raspberry Pi and interfaced with an RPLidar in the front top portion of the bot. This low-cost mapping bot has features such as SLAM (Simultaneous Localization and Mapping), which can not only form a map of the environment using Lidar scans and Frame Grabber's Robotic Operating System Software package to communicate with ROS in the Raspberry Pi via ROS Network Configurations, but also additional functions such as requesting the user to select the destination, path planning to safely reach the destination, and then gathering the required small resources / samples from the hazards underground areas. The project's goal is to create a fully autonomous robot capable of mapping its surroundings and navigating obstacles. This can be accomplished by utilising a chassis outfitted with tracks and two motors, a Lidar, a compass, and a Raspberry Pi. The work done & presented in this paper is the result of the final year one year project work that has been done by the final year engineering students of the college and as such there is little novelty in it and the references are being taken from various sources from the internet, the paper is being written by the students to test their writing skills in the final stages of their engineering career and also to test the presentation skills during their final year project presentation and the work done & presented in this paper is the report of the undergraduate project work done by the students.

Keywords Four-wheeled, Lidar bot, globetrotter, terrain bot, autonomous bot.

1. Introduction

The Turtle bot is a widely recognized product that utilizes advanced technologies such as SLAM (Simultaneous Localization and Mapping) and Navigation, making it highly suitable for home service robots. This versatile bot comes in various versions, each offering unique structures and features. Equipped with gyro, kinetic sensors, Lidar, and a laptop with Mapping capabilities [3], it employs ROS (Robot Operating System) to facilitate localization and navigation on a Raspberry Pi. Communication between the bot and the Frame Grabber is achieved through ROS software packages. ROS is an open-source software library collection used for developing robotic applications, encompassing algorithms and development tools [2]. Its compatibility with different operating systems allows seamless integration with the Frame Grabber, enabling access to the ROS bot. SLAM, which stands for Simultaneous Localization and Mapping, can be illustrated using the analogy of Google Maps, where the entire world is mapped out. SLAM's mapping process is akin to that concept [4].

Likewise, GPS (Global Positioning System) enables individuals to be located on a map using latitude and longitude coordinates [5]. Another example involves the use of Lidar, which allows for environment mapping. By employing the SLAM algorithm, both localization and mapping can be performed simultaneously. Lidar, short for Light Detection and Ranging, employs pulsed laser light to detect obstacles within its range. By continuously obtaining ranges at precise angles throughout the environment, a comprehensive map can be generated. This map, known as laser scans, includes information such as ranges, angles, Cartesian coordinates, and the corresponding values [1]. LIDAR has various classifications based on physical and scattering processes and the platform it operates on. One such robot is the Globetrotter, which utilizes the ROS (Robotic Operating System) and interfaces a Raspberry Pi with an RPLidar sensor [6]. This robot boasts features like SLAM, allowing it to map the environment and perform additional functions such as requesting user input for destination selection, path planning to reach the destination while avoiding obstacles, and gathering necessary resources or samples from hazardous underground areas. Refer to Figure 1 for a visual representation of the developed model [7].

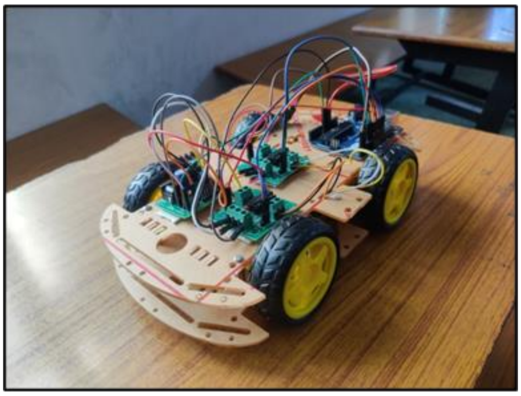

Fig. 1 : Developed model of robo

2. Literature Survey

A no. of researchers have worked on the topic of the project works put in this papers. Here, follows a brief review of the works done by the various authors.

[1] "A Survey of Autonomous Navigation Techniques for Mobile Robots" by Shih-Chung Kang and Yu-Wei Chen - This paper provides an overview of various navigation techniques used in mobile robots, including reactive, deliberative, and hybrid approaches.

[2] "Robot Navigation: A Survey" by Jana Tumova and David Obdrzalek - This article provides a comprehensive survey of robot navigation techniques, including those based on laser range finders, vision, and sonar.

[3] "Navigation for Autonomous Mobile Robots" by Dieter Fox - This book provides an in-depth review of navigation algorithms and techniques for autonomous mobile robots, including topics such as sensor fusion, localization, mapping, and path planning.

[4] "Vision-Based Navigation for Mobile Robots: A Survey" by C. K. Liu and K. M. Lee - This paper presents an overview of vision-based navigation techniques used in mobile robots, including visual odometry, visual simultaneous localization and mapping (SLAM), and visual servoing.

[5] "Development of a Mobile Robot for Rough Terrain Exploration" by Hyun-Taek Choi and Jin-Woo Jung - This article describes the design and development of a mobile robot capable of navigating rough terrain using a combination of sensors and control algorithms.

[6] "Terrain-adaptive Locomotion of Legged Robots" by Sangbae Kim and Matthew Travers - This paper presents a method for adaptive locomotion of legged robots on varying terrain, which enables the robot to adjust its gait and posture in response to changing terrain conditions.

[7] "Robust Terrain Classification for Autonomous Off-Road Driving" by Michael Dixon, Paul Newman, and Ingmar Posner - This paper describes a method for robust terrain classification using machine learning techniques, which is used to improve the performance of autonomous off-road driving systems.

[8] "A Survey of Artificial Intelligence Techniques for Navigation in Robotics" by R. M. Menezes, G. A. Gauthier, and P. B. de Moura Oliveira - This paper provides an overview of various artificial intelligence techniques used in robot navigation, including fuzzy logic, neural networks, and genetic algorithms.

[9] "Autonomous Navigation for Unmanned Aerial Vehicles: A Survey" by C. N. Nwokah, Y. J. Chang, and K. J. Chung - This paper presents a survey of autonomous navigation techniques for unmanned aerial vehicles (UAVs), including topics such as perception, motion planning, and control.

[10] "A Survey of SLAM Algorithms for Autonomous Navigation" by Xuesong Shi and Xiaodong Xu - This paper provides a comprehensive survey of simultaneous localization and mapping (SLAM) algorithms used in autonomous navigation, including visual, lidar, and fusion-based approaches.

3. Scope and objectives

The objective of this research paper is to evaluate the existing literature on lane sensing and tracing for ADAS (Advanced Driver Assistance Systems) and identify knowledge gaps for future experimentation [8]. The paper compiles and compares various lane sensing and tracing algorithms while examining the unique datasets utilized for algorithm validation, along with the metrics employed for evaluation. The review categorizes current lane recognition and tracing algorithms into three groups: function-based, mastery-based, and model-based, offering a structured approach to understanding the fundamental aspects of these algorithms in the literature. Additionally, it compares select copyrighted works by automotive manufacturers with prominent commercialization efforts in this field, although conducting a comprehensive analysis of patented works is challenging due to the large number of patents issued by educational institutions, research organizations, and vehicle manufacturers [9].

This systematic review aims to assist researchers in this field by highlighting the latest advancements in lane identification and tracking for ADAS, as well as the challenges that need to be addressed for reliable lane sensing and monitoring systems in the future. The outline of this review paper is as follows: The second section provides an overview of the methodology used to review the literature. This is followed by an in-depth analysis of the literature, which includes a detailed explanation of the sensors employed in ADAS and a comprehensive review of the current literature on lane sensing and monitoring algorithms. The deliberations in the fourth section are followed by the findings presented in Figure 2 in the fifth section [10].

4. Proposed Methodology

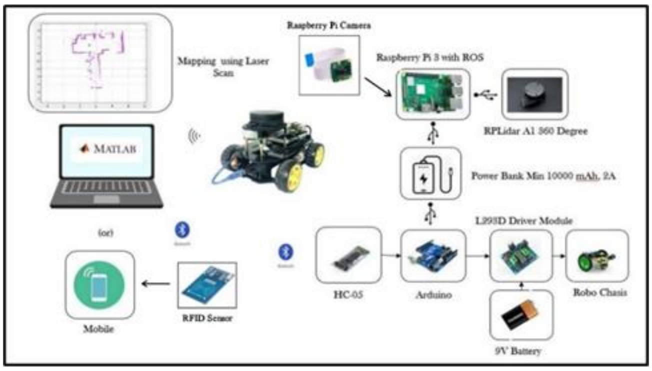

Fig. 2 : Proposed methodology of the block diagram

5. Assembling of the Robot’s Frames Structure

For this project, the decision was made to utilize a continuous track, commonly known as a tank tread, in order to maximize traction and ensure precise movements. Each track is driven by its own motor and encoder, providing independent control. To optimize space efficiency, two separate platforms were created for mounting components, with the motors positioned underneath. The lower platform accommodates the Raspberry Pi, two LiPo batteries, two voltage regulators, two electronic speed controllers, the BNO055 absolute orientation sensor, and a breadboard. On the upper platform, the LiDAR is mounted independently to ensure an unobstructed scanning plane [11]. To prevent shifting during robot rotation, the LiDAR and compass have been centrally mounted on the chassis. Additionally, a symmetrical mounting arrangement has been implemented to maintain balance and prevent veering during movement. The LiPo batteries utilized have a capacity of 7.4V 2200mAh. Both voltage regulators are employed to step down the voltage from 5V to 5V. One battery powers the Raspberry Pi, which, in turn, supplies power to the LiDAR and absolute orientation sensor. The second battery powers the motors [12].

6. LiDARs

In our setup, the RPLIDAR A1M8 communicates with the Raspberry Pi via a UART to serial USB converter. Leveraging the USB interface functionality provided by the Raspbian operating system, using a Raspberry Pi allows us to easily interact with serial devices. The communication process involves opening, reading, and writing to the file handle /dev/ttyUSB0, which acts as a buffer for the LiDAR [13]. The LiDAR supports various commands that enable us to initiate and terminate a scan, retrieve health information, and restart the device. During scanning, the LiDAR transmits multiple data packets, each containing distance, angle, and scan quality information. These packets come in different formats, and to efficiently handle them when reading and writing data, we have created a C struct for each packet format. To control the rotation of the LiDAR motor, we power it on and off by enabling and disabling the USB DTR (Data Terminal Ready) control signal [14].

7. Mappings

Information about the robot’s surroundings is stored in a two dimensional array that will serve as a grid. In this project, we utilize a two-dimensional array as a grid to store information about the robot's surroundings. Each index in the array is represented by a single byte, allowing us to designate it as empty, an obstacle, or not yet scanned, denoted by the value 6. Each index corresponds to a square space in the physical environment of the robot. The size of each square in the index is determined by a scale defined in a header file, which can be easily adjusted. A smaller scale provides higher resolution maps. All distance measurements in this project are in millimeters. For our purposes, a scale of 20mm offers sufficient resolution for movement and detecting small objects.

The overall size of the map is also flexible and adjustable through a C definition [15]. The two-dimensional array representing the entire map is declared as map[MAP_SIZE][MAP_SIZE]. The value of MAP_SIZE is determined as ten times the double of the maximum range that the LiDAR can scan, divided by the scale of each index. For instance, if the LiDAR has a maximum range of 24000 units and our scale is 20, then the calculation would be 10 * (24000 / 20). This allows us to create a map with an area of 57,600 square meters while utilizing only around 10% of the available memory. In theory, by increasing the multiple to 25, we could use almost all of the RAM memory and generate a map with an area of 360,000 square meters. The LiDAR range is doubled to account for the fact that we initialize the robot's location at the center of the map. Indexing into the two-dimensional array follows an X, Y convention (map[y][x]), where (0,0) represents the farthest north-west index, (max, max) represents the farthest south-east index, and (max/2, max/2) represents the center of the map [16].

8. Motor

To facilitate movement, we employ two brushless DC motors with a 1:75 gear ratio, providing ample torque to drive the chassis. These motors are equipped with hall effect encoders, enabling us to calculate the linear distance traveled. For seamless control over speed and direction, we utilize two TI DRV8838 electronic speed controllers [17]. By supplying a pulse width modulation (PWM) signal to the DRV8838, we can adjust the proportional DC output voltage sent to the motors, thereby controlling their speed. The DRV8838 also features a Phase input that determines the motor's rotation direction. The Raspberry Pi generates two individual PWM signals at a frequency of 1.92kHz, one for each electronic speed controller [17].

Due to the PWM signal generation of the Raspberry Pi, the relatively low frequency allows for a duty cycle resolution of 100 steps. This means that the duty cycle can be varied in increments of 1%, providing a range from 0% to 100% with a 100-step resolution. Consequently, this variable duty cycle PWM signal ensures a flexible and precise control over the output voltage of the DRV8838, maintaining the 100-step resolution. For example, if the duty cycle is set to 75%, the DRV8838 will output a DC voltage equivalent to 75% of the motor voltage supply (VIN, pin 7, on the DRV8838). Additionally, the Raspberry Pi is configured to provide two individual GPIO outputs, which are connected to the phase inputs of the DRV8838, allowing each motor to operate independently. This setup enables accurate calibration of movement and turning. To track linear distance, the hall sensors on the motors trigger interrupts on the Raspberry Pi. For precise turns, we rely on the Bosch BNO055 absolute orientation sensor [18].

9. Orientations Sensors

To ensure constant tracking of the robot's orientation, we opted to utilize the Bosch BNO055 absolute orientation sensor. Adafruit has integrated this sensor onto a breakout board and provided a driver library that communicates over UART for Raspberry Pi usage. However, since our project is implemented in C and the driver library is written in Python, we needed to create a wrapper file to embed the Python module in C [19]. For proper functioning, the orientation sensor requires manual calibration initially. This involves continuously moving and rotating the sensor for a brief period of time. The calibration settings can be saved and loaded onto the sensor each time it is powered on. Our system has been designed to establish the initial direction the robot faces as its relative North upon power-on, as the orientation sensor does not always provide an absolute North. The robot's facing angle ranges from 0 to 360 degrees in a clockwise fashion [20].

10. Path Planning

In order to determine the optimal path from point A to B while avoiding obstacles, we employ the A-Star algorithm [21]. This algorithm is an informed search algorithm that assesses the cost of each movement to determine the most suitable path. The cost of moving to a specific location is calculated by summing the number of movements made to reach the current position and a heuristic that estimates the shortest remaining distance to the destination. In our implementation, the heuristic is computed as the absolute difference between the current x-coordinate and the end x-coordinate, plus the absolute difference between the current y-coordinate and the end y-coordinate. Prior to considering a movement to the next location, the algorithm checks if that particular index on the map contains an obstacle [22].

Currently, the algorithm has been designed to enable horizontal and vertical movements on the map. However, it can be further enhanced to account for diagonal movements in the future. The algorithm maintains two lists: one for tracking all possible next movements and another for recording previously attempted movements. These lists are managed using a binary minimum heap and a regular array as the underlying data structures. The efficiency of the algorithm in finding a path depends greatly on the performance of these data structures in adding new elements, searching for specific elements, and identifying the element with the lowest cost [23].

11. Final Designs Concepts

The entire system is designed to facilitate the integration of all components and peripherals, allowing for easy replacement of individual devices if needed. In our final system design, we initiate the mapping process by populating the map with information obtained from the LiDAR scans [28]. To achieve this, we first populate a smaller grid that will later be merged into the overall map. This approach enables potential implementation of a multithreaded system in the future. The initial grid size is determined to accommodate the immediate environment within the LiDAR's range.

To populate the grid, we request a single 360-degree scan from the LiDAR [27]. By utilizing the distance, angle, and the current location of the robot (which is initially set as the center of the smaller grid), we can apply basic trigonometry to determine the precise location of each object in the robot's environment. The calculated angle incorporates the angle provided by the LiDAR and the direction of the robot as measured by the compass, ensuring accurate positioning relative to the robot [26].

To account for moving objects, we mark each index between the scanned object and the robot as empty. With the robot's actual location on the map, we update that specific section of the map with the newly populated smaller grid. Each time the map is updated, the robot attempts to calculate a path from its current location to the user-defined destination [25]. If a viable path is found, a list of x,y coordinates representing the path is returned. This list is then translated into a sequence of rotations and forward movements. Utilizing the current direction of the robot obtained from the compass and the current and next ten indexes it needs to traverse, we determine the required rotation [24]. The motors are then controlled to execute the necessary rotation, followed by moving forward the distance specified by our chosen scale.

In scenarios where the robot needs to navigate around moving obstacles, it updates the map after each individual movement. However, when operating in an environment with stationary obstacles, the robot can continue processing movements until it reaches an unscanned index on the map [29]..

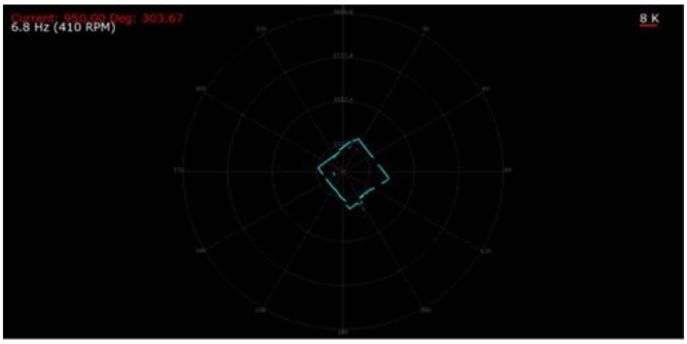

Fig. 3 : Simulation of the mapping of the robot for path planning

12. Conclusion

The objective of this project was to demonstrate the utilization of LiDAR for autonomous mapping and navigation of an area. The system developed in this project can serve as a foundational model for implementing commercial equipment such as autonomous vacuums or forklifts. Despite using lower quality components in the current design, we were still able to achieve highly accurate maps and navigate the robot within the mapped area. Our future plans include implementing enhanced mobility for the robot, allowing for more fluid movement beyond just 90-degree turns. Additionally, we aim to make the system multithreaded to reduce the scanning and path calculation time. Overall, we believe that this project has successfully brought together the knowledge and skills we have acquired throughout our computer engineering studies, and we are pleased with the outcomes achieved..

References

[1] Badue, C., Guidolini, R., Carneiro, R. V., Azevedo, P., Cardoso, V. B., Forechi, A., ... & Veronese, L. (2019). Self-driving cars: A survey. arXiv preprint arXiv:1901.04407.

[2] Khairuddin, A. R., Talib, M. S., &Haron, H. (2015, November). Review on simultaneous localization and mapping (SLAM). In 2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE) (pp. 85-90). IEEE.

[3] Feder, H., Leonard, J., & Smith, C. (1999). Adaptive Mobile Robot Navigation and Mapping. The International Journal Of Robotics Research, 18(7), 650-668

[4] Leonard, J., &Durrant-Whyte, H. (1992). Directed sonar sensing for mobile robot navigation. Boston: Kluwer Academic Publishers.

[5] Leonard, J., & Feder, H. (2001). Decoupled stochastic mapping [for mobile robot & AUV navigation]. IEEE Journal Of Oceanic Engineering, 26(4), 561-571

[6] Chong, T., Tang, X., Leng, C., Yogeswaran, M., Ng, O., & Chong, Y. (2015). Sensor Technologies and Simultaneous Localization and Mapping (SLAM). Procedia Computer Science, 76, 174-179.

[7] Bresson, G., Alsayed, Z., Yu, L., & Glaser, S. (2017). Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. IEEE Transactions On Intelligent Vehicles, 2(3), 194-220.

[8] Grisetti, G., Stachniss, C., &Burgard, W. (2007). Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. IEEE Transactions On Robotics, 23(1), 34-46. doi: 10.1109/tro.2006.889486

[9] Cheng, H.-Y.; Yu, C.-C.; Tseng, C.-C.; Fan, K.-C.; Hwang, J.-N.; Jeng, B.-S. Environment classification and hierarchical lane detection for structured and unstructured roads. Comput. Vis. IET 2010, 4, 37–49.

[10] Han, J.; Kim, D.; Lee, M.; Sunwoo, M. Road boundary detection and tracking for structured and unstructured roads using a 2D lidar sensor. Int. J. Automot. Technol. 2014, 15, 611–623.

[11] Le, M.C.; Phung, S.L.; Bouzerdoum, A. Lane Detection in Unstructured Environments for Autonomous Navigation Systems. In Asian Conference on Computer Vision; Cremers, D., Reid, I., Saito, H., Yang, M.H., Eds.; Springer: Cham, Switzerland, 2015.

[12] Wang, J.; Ma, H.; Zhang, X.; Liu, X. Detection of Lane Lines on Both Sides of Road Based on Monocular Camera. In Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China, 5–8 August 2018; pp. 1134–1139.

[13] YenIaydin, Y.; Schmidt, K.W. Sensor Fusion of a Camera and 2D LIDAR for Lane Detection. In Proceedings of the 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 24–26 April 2019; pp. 1–4.

[14] Dr. T.C. Manjunath, Pavithra G., “An Optimized Method Using CNN, RF, Cuckoo Search and HOG for Early Detection of Eye Disease in Humans”, European Journal of Electrical Engineering and Computer Science, EJECE, Belgium, Impact Factor 1.89, ISSN 2506-9853, Vol. 4, No. 2, pp. 1-4, 13 April 2020.

[15] Pavithra G., Dr. T. C. Manjunath, “Use of Artificial Intelligence & Machine Learning with Deep Learning for Glaucoma Detection in Human Eyes & its Real Time Hardware Implementation”, European Journal of Electrical Engineering and Computer Science, EJECE, Belgium, Impact Factor 1.89, ISSN 2506-9853, Vol. 4, No. 2, pp. 5-10, 30 April 2020.

[16] Pavithra G., Dr. T. C. Manjunath, “Early detection of eye disease in humans using Random Forest & HOG concepts”, SSRG International Journal of Electronics and Communication Engineering (IJECE), E-ISSN 2348 – 8549, P-ISSN 2349 – 9184, Seventh Sense Research Group® (SSRG), Impact Factor 2.03, Paper id IJECE-V7I4P102, Vol. 7, Issue 4, pp. 5 – 7, April 2020.

[17] Pavithra G., Dr. T. C. Manjunath, “AI, ML and the Eye Disease Detection”, SSRG International Journal of Computer Science and Engineering (IJCSE), P-ISSN 2348 - 8387, Seventh Sense Research Group® (SSRG), Impact Factor 2.15, Paper id IJCSE-V7I4P101, Vol. 7, Issue 4, pp. 1 – 3, April 2020.

[18] Pavithra G., Dr, T. C. Manjunath, “A Review on HOG Feature Extraction based LDA Classification on Medical Image Processing”, International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056, p-ISSN: 2395-0072, Paper Id 804, Impact Factor value: 7.529, ISO 9001:2008 Certified Journal Vol. 7, Issue 4, pp. 4192-4193, Apr. 2020.

[19] Dr. T.C.Manjunath, Dr. K.N.Vijaykumar, Pavithra G., “Separation of foreground & background objects in image processing”, National Journal of Applied Engineering and Technologies (JAET), ISSN-2278-1722, Paper id AET-0014, Vol. 1, Issue 1, pp. 60 - 65, April 2012, India.

[20] Dr. T.C.Manjunath, Pavithra G., Suhasini V.K., “Lossless compression in artificial images”, National Journal of Applied Engineering and Technologies (JAET), ISSN-2278-1722, Paper id AET-0015, Vol. 1, Issue 1, pp. 66 - 71, April 2012, India.

[21] Dr. T.C.Manjunath, Dr. Vaibhav Meshram, Pavithra G., “Dynamic modelling of a robot arm”, Journal of Applied Engineering and Technologies (JAET), ISSN-2278-1722, Paper id AET-012, Vol. 2, Issue 1, pp. 47 - 53, April 2013, India.

[22] Dr. T.C.Manjunath, Arunkumar G., Pavithra G., “Development of swarm intelligence in mobile robotic systems”, Journal of Applied Engg. & Technologies (AET-2014), ISSN : 2278 – 1722, Vol. 3, Issue 1, Apr. 2014, pp. 109-112, Vidyalankar Inst. of Tech., Mumbai, Maharashtra, India.

[23] Kemsaram, N.; Das, A.; Dubbelman, G. An Integrated Framework for Autonomous Driving: Object Detection, Lane Detection, and Free Space Detection. In Proceedings of the 2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4), London, UK, 30–31 July 2019; pp. 260–265.

[24] Lee, C.; Moon, J.-H. Robust Lane Detection and Tracking for Real-Time Applications. IEEE Trans. Intell. Transp. Syst. 2018, 19, 1–6.

[25] Son, Y.; Lee, E.S.; Kum, D. Robust multi-lane detection and tracking using adaptive threshold and lane classification. Mach. Vis. Appl. 2018, 30, 111–124

[26] Li, Q.; Zhou, J.; Li, B.; Guo, Y.; Xiao, J. Robust Lane-Detection Method for Low-Speed Environments. Sensors 2018, 18, 4274.

[27] Son, J.; Yoo, H.; Kim, S.; Sohn, K. Real-time illumination invariant lane detection for lane departure warning system. Expert Syst. Appl. 2014, 42.

[28] Chae, H.; Jeong, Y.; Kim, S.; Lee, H.; Park, J.; Yi, K. Design and Vehicle Implementation of Autonomous Lane Change Algorithm based on Probabilistic Prediction. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2845–2852.

[29] Chen, P.R.; Lo, S.Y.; Hang, H.M.; Chan, S.W.; Lin, J.J. Efficient Road Lane Marking Detection with Deep Learning. In Proceedings of the 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), Shanghai, China, 19–21 November 2018; pp. 1–5.