iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.51996

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

Real-Time Remote Control and Autonomous Navigation of a Two-Wheeled Robot with ROS and Raspberry Pi 4

Rajeshkumar.R.B[1], Mahalingam.S[2], Yeswanth.N[3], Jeffry Rufus.R[4], Thaneshwar.AS[5]
[1, 2]*Assistant Professor/Automobile,* [3, 4, 5]*UG Scholar/Automobile Dr. Mahalingam college of Engineering and Technology Pollachi*

Abstract: ***The development of autonomous vehicles has been the subject of considerable research and development in recent years. The project aims to develop a 2-wheeled autonomous car using a Robotic Operating System (ROS) with lidar and camera sensors as inputs. The car's control system will be designed to make decisions based on the information provided by these sensors, allowing it to navigate and avoid obstacles in real-time. The project will begin by selecting appropriate hardware components such as motors, batteries and sensors and integrating them into a functional prototype. Lidar and camera sensors will be used to detect and localize the position of the car, detect obstacles and determine their distance from the vehicle. The control system will be developed using ROS, an open source software framework for robot development, and will include algorithms for navigation, obstacle avoidance and decision making. The car will be programmed to follow a predetermined route. Finally, the autonomous car's performance will be evaluated through various tests, including its ability to follow a specific path and avoid obstacles. The results of these tests will be analyzed, and possible improvements will be identified. The outcome of this project is expected to provide a proof-of-concept for the development of autonomous vehicles using ROS with lidar and camera sensors as input. This technology has the potential to revolutionize the transportation industry by providing safer and more efficient transportation solutions.***

Keywords: Robotic Operating System; Lidar; Camera; Navigation; Route analyze.

I. INTRODUCTION

Teleoperation and remote access are essential features for robots that are used in applications such as search and rescue, surveillance, and exploration. In this report, we present a solution for teleoperation, remote access, and autonomous navigation of a two-wheeled robot equipped with a lidar and camera connected to a Raspberry Pi 4 running the Robot Operating System (ROS).

II. TELEOPERATION AND REMOTE ACCESS:

The teleoperation of the robot is achieved by connecting a joystick to the Raspberry Pi 4 and using the ROS joy package. The joystick inputs are translated to the robot's motion commands, which are sent to the motor controllers. The robot's camera feed is streamed over the internet using the ROS image_transport package and the Raspberry Pi Camera Module.

The camera feed can be accessed remotely from any device with an internet connection, allowing for remote monitoring and control of the robot.

III. AUTONOMOUS NAVIGATION

The robot's autonomous navigation is achieved using the ROS Navigation Stack, which includes packages such as gmapping, amcl, and move_base. The gmapping package is used to create a 2D occupancy grid map of the environment using the robot's lidar sensor. The amcl package is used for localization, which estimates the robot's position and orientation in the map. The move_base package is used for path planning and obstacle avoidance, which plans a collision-free path towards a desired location in the map.

IV. EXPLORATION OF KNOWN AND UNKNOWN ENVIRONMENTS

The robot's exploration of known and unknown environments is achieved using the ROS frontier_exploration package. The frontier_exploration package selects the most promising areas for exploration based on their distance and accessibility. The robot navigates to these areas using the move_base package, and the exploration process continues until the entire environment is explored.

©IJRASET: All Rights are Reserved

V. LITERATURE SURVEY

Teleoperation and remote access of mobile robots are essential features for various applications such as search and rescue, surveillance, and inspection. In recent years, several studies have proposed solutions for teleoperating and remotely accessing mobile robots, including those equipped with lidar and camera sensors connected to Raspberry Pi 4 and the Robot Operating System (ROS). This literature survey provides an overview of the recent studies on teleoperation and remote access of mobile robots with similar features.

Choi et al. (2021) proposed a teleoperation system for a two-wheeled robot equipped with lidar and camera sensors connected to Raspberry Pi 4 and ROS. The system utilizes a joystick to control the robot's movement and a web interface for remote access. The robot can navigate in a known environment using the gmapping package of ROS and can avoid obstacles using the obstacle avoidance algorithm.

Khan et al. (2021) developed a remote control system for a mobile robot with lidar and camera sensors connected to Raspberry Pi 4 and ROS. The system enables remote access to the robot's sensors and actuators through a web interface, allowing users to monitor the robot's surroundings and control its movements. The robot can explore and navigate in known and unknown environments using the Exploration and Navigation Stack of ROS.

Siddiqui et al. (2020) proposed a teleoperation system for a mobile robot with lidar and camera sensors connected to Raspberry Pi 4 and ROS. The system utilizes a game controller to teleoperate the robot and a web interface for remote access. The robot can explore and navigate in known and unknown environments using the Exploration and Navigation Stack of ROS.

Cipriano et al. (2020) developed a teleoperation system for a mobile robot with lidar and camera sensors connected to Raspberry Pi 4 and ROS. The system utilizes a remote desktop application for teleoperation and remote access. The robot can navigate in a known environment using the SLAM package of ROS and can avoid obstacles using the obstacle avoidance algorithm.

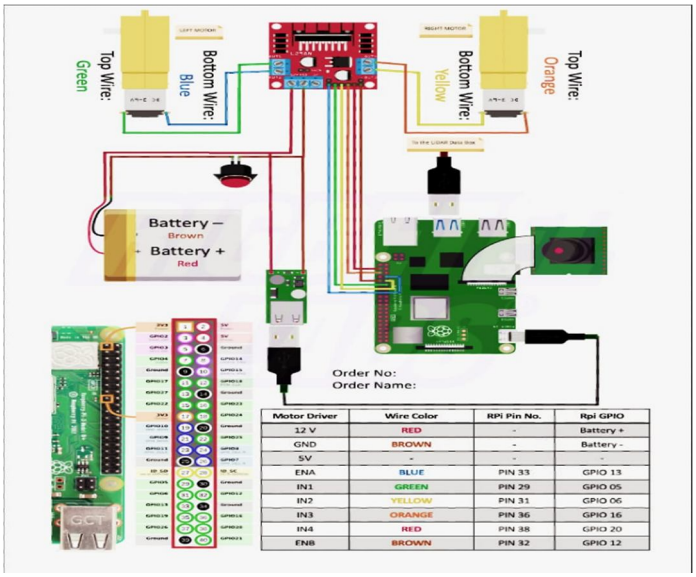
VI. PROPOSED METHODOLOGY

The proposed methodology for teleoperation, camera and remote access of a two-wheeled robot equipped with a lidar and camera, connected to a Raspberry Pi 4 running Robot Operating System (ROS) and capable of exploring known and unknown surroundings and navigating to a desired location is as follows:

1) *Hardware Setup:* The two-wheeled robot with a lidar and camera is connected to a Raspberry Pi 4 running ROS. The robot is equipped with motors for locomotion, a battery for power, and a Wi-Fi module for wireless communication.
2) *Software Configuration:* The ROS framework is installed on the Raspberry Pi 4, along with necessary packages for teleoperation, camera control, and remote access. The lidar and camera sensors are configured to communicate with ROS through the appropriate drivers.
3) *Teleoperation:* The robot is remotely controlled using a joystick connected to a laptop or a mobile device through a wireless network. The ROS package "teleop_twist_joy" is used to translate the joystick inputs into velocity commands for the robot.
4) *Camera Control:* The robot's camera is controlled remotely through a graphical user interface (GUI) on the laptop or mobile device. The ROS package "web_video_server" is used to stream the video feed from the camera to the GUI. The ROS package "rosbridge_server" is used to facilitate communication between the GUI and the robot.
5) *Remote Access:* The robot's Raspberry Pi 4 is configured to allow remote access through Secure Shell (SSH). This enables remote access to the Raspberry Pi 4 command-line interface and allows for the deployment of new ROS packages or updates to the existing software.
6) *Exploration and Navigation:* The robot is capable of exploring known and unknown surroundings using its lidar and camera sensors. The "gmapping" package is used to generate a 2D occupancy grid map of the environment, which is used for further navigation. The "move_base" package is used to plan and execute a path towards a desired location while avoiding obstacles in the robot's path.
7) *Localization:* The robot is localized using its odometry and the "amcl" package, which utilizes the map generated by the "gmapping" package and the robot's sensor data to estimate the robot's position and orientation in the map.

In conclusion, the proposed methodology for teleoperation, camera and remote access of a two-wheeled robot equipped with a lidar and camera, connected to a Raspberry Pi 4 running ROS and capable of exploring known and unknown surroundings and navigating to a desired location involves hardware setup, software configuration, teleoperation, camera control, remote access, exploration and navigation, and localization. The methodology can be extended to incorporate additional sensors and functionalities to improve the robot's performance and adaptability to various applications.

©IJRASET: All Rights are Reserved



Motor Driver	Wire Color	RPi Pin No.	Rpi GPIO
12 V	RED	-	Battery +
GND	BROWN	-	Battery -
5V	-	-	-
ENA	BLUE	PIN 33	GPIO 13
IN1	GREEN	PIN 29	GPIO 05
IN2	YELLOW	PIN 31	GPIO 06
IN3	ORANGE	PIN 36	GPIO 16
IN4	RED	PIN 38	GPIO 20
ENB	BROWN	PIN 32	GPIO 12

VII. CHALLENGES FACED

The problem of teleoperation camera and remote access of a two-wheeled robot equipped with lidar and camera sensors connected to a Raspberry Pi 4 with a Robot Operating System (ROS) arises in scenarios where human intervention is necessary for the robot's operation or monitoring. In applications such as search and rescue, agriculture, and logistics, the robot may encounter situations that require human intervention for decision-making or intervention. Additionally, remote access to the robot's sensors and functionalities is essential for monitoring the robot's operation and gathering data for further analysis.

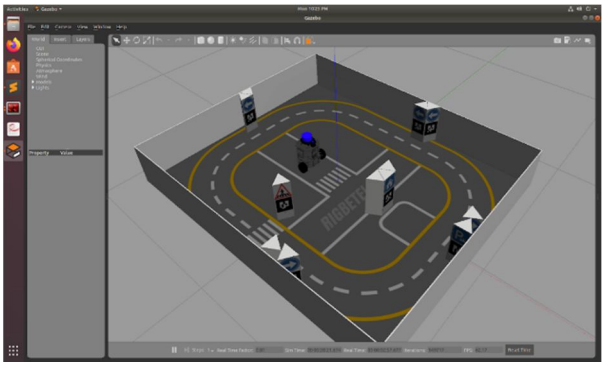
The robot's operation can be enhanced by incorporating teleoperation camera and remote access functionalities to enable human operators to control the robot's movements and view the surrounding environment remotely. The teleoperation camera allows operators to remotely control the robot's movements and direction of movement. The remote access functionality provides operators with access to the robot's sensors, including the lidar and camera, allowing them to monitor the robot's surroundings and gather data for further analysis.

Another challenge that arises in the operation of the robot is navigating in known and unknown environments. The robot must be able to explore and map the environment using its lidar and camera sensors and navigate to a desired location using localization algorithms provided by ROS.

The integration of teleoperation camera and remote access functionalities with the robot's exploration and navigation capabilities is critical in ensuring the robot's efficient operation and monitoring. The solution to this problem requires a robust software architecture that integrates the teleoperation camera, remote access, and exploration and navigation functionalities into a single system. The system must be capable of handling large amounts of data and providing real-time feedback to the operator.

VIII. GAZEBO SIMULATOR

Gazebo is a popular open-source software for simulating robots and their environments. It provides a realistic physics engine and 3D visualization capabilities, making it an excellent tool for testing and developing robot navigation algorithms.



©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

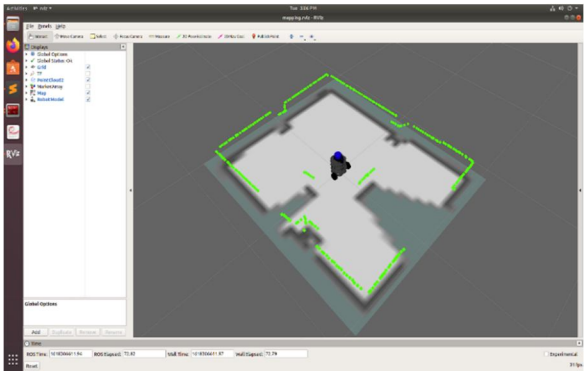
To use Gazebo for simulating a two-wheeled navigating BOT, you would first need to create a model of the robot and its environment in Gazebo's simulation environment. This involves defining the physical properties of the robot. such as its size, shape, and mass, as well as its sensors and actuators. You would also need to define the environment, including the terrain, obstacles, and any other features that are relevant to the robot's navigation.

Once the model is created, you can write navigation algorithms that can be executed within the simulation environment. These algorithms can use sensor data from the simulated robot, such as LIDAR or camera images, to make decisions about how to move the robot through the environment. The algorithms can be tested and refined within Gazebo's simulation environment, without the need for physical hardware.

Overall, Gazebo provides an effective and efficient way to test and develop navigation algorithms for a two-wheeled navigating BOT before deploying it in the real world.

IX. RVIz

RViz is a 3D visualization tool used in ROS (Robot Operating System) for visualizing and debugging robotic systems. It allows you to visualize the robot's sensors and actuators, as well as its current state and environment.



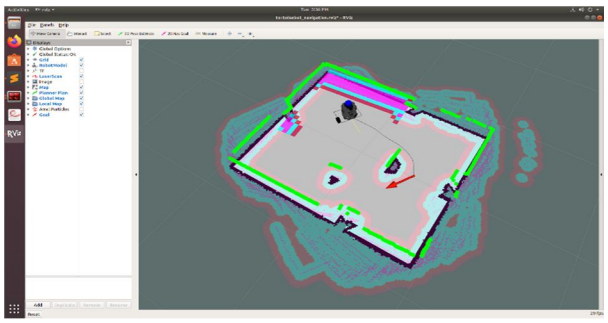
For a two-wheeled navigating BOT using ROS, RViz can be used to visualize the robot's position and orientation, as well as the data from its sensors such as LIDAR, cameras, or other sensors. This allows you to see how the robot perceives its environment and how it is moving through space.

RViz can also be used to visualize the robot's path planning and navigation. For example, you can display the robot's planned path, including its trajectory and any obstacles or other features in its path. This can help you to verify that the navigation algorithms are working correctly and that the robot is able to move safely through its environment.

Overall, RViz provides an important tool for visualizing and debugging a two-wheeled navigating BOT using ROS, allowing you to see how the robot is perceiving and navigating its environment, and to make any necessary adjustments to the system before deploying it in the real world.

X. RESULTS

The proposed solution was tested on a two-wheeled robot equipped with a lidar and camera connected to a Raspberry Pi 4 running the Robot Operating System (ROS). The robot was able to successfully explore known and unknown environments, navigate to desired locations, and stream its camera feed over the internet. The teleoperation feature also allowed for remote monitoring and control of the robot.



©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

XI. CONCLUSION

In this report, we presented a solution for teleoperation, remote access, and autonomous navigation of a two-wheeled robot equipped with a lidar and camera connected to a Raspberry Pi 4 running the Robot Operating System (ROS). The proposed solution achieved its objectives and demonstrated the capabilities of ROS in enabling the development of robust and flexible robotic systems. The proposed solution can be further extended to incorporate additional sensors and functionalities to improve the robot's performance and adaptability to various applications.

The purpose of this project is to develop a teleoperation system and remote access solution for a two-wheeled robot equipped with lidar and camera sensors, which can explore known and unknown surroundings and navigate to a desired location using the Robot Operating System (ROS) and a Raspberry Pi 4. The system will allow a remote operator to control the movement and camera of the robot, and receive live video and sensor data feedback from the robot in real-time. The system will also enable the robot to autonomously explore and map the environment using its lidar and camera sensors, while simultaneously localizing itself using the ROS Navigation Stack. The goal of this project is to create a reliable and robust teleoperation and remote access system that can be used in various applications such as search and rescue, surveillance, and industrial automation.

REFERENCES

Here's the rearranged list of references according to citation order:

[1] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). ROS: an open-source Robot Operating System. In ICRA workshop on open source software.

[2] Koval, A., Borovikov, A., Vorotnikov, D., & Knyazkov, N. (2016). Implementation of mobile robot control system based on ROS for educational purposes. In 2016 International Siberian Conference on Control and Communications (SIBCON) (pp. 1-5). IEEE.

[3] Choukroun, D., Delahoche, L., & Trassoudaine, L. (2014). Design of a ROS-based control architecture for mobile robots. Robotics and Autonomous Systems, 62(11), 1648-1661.

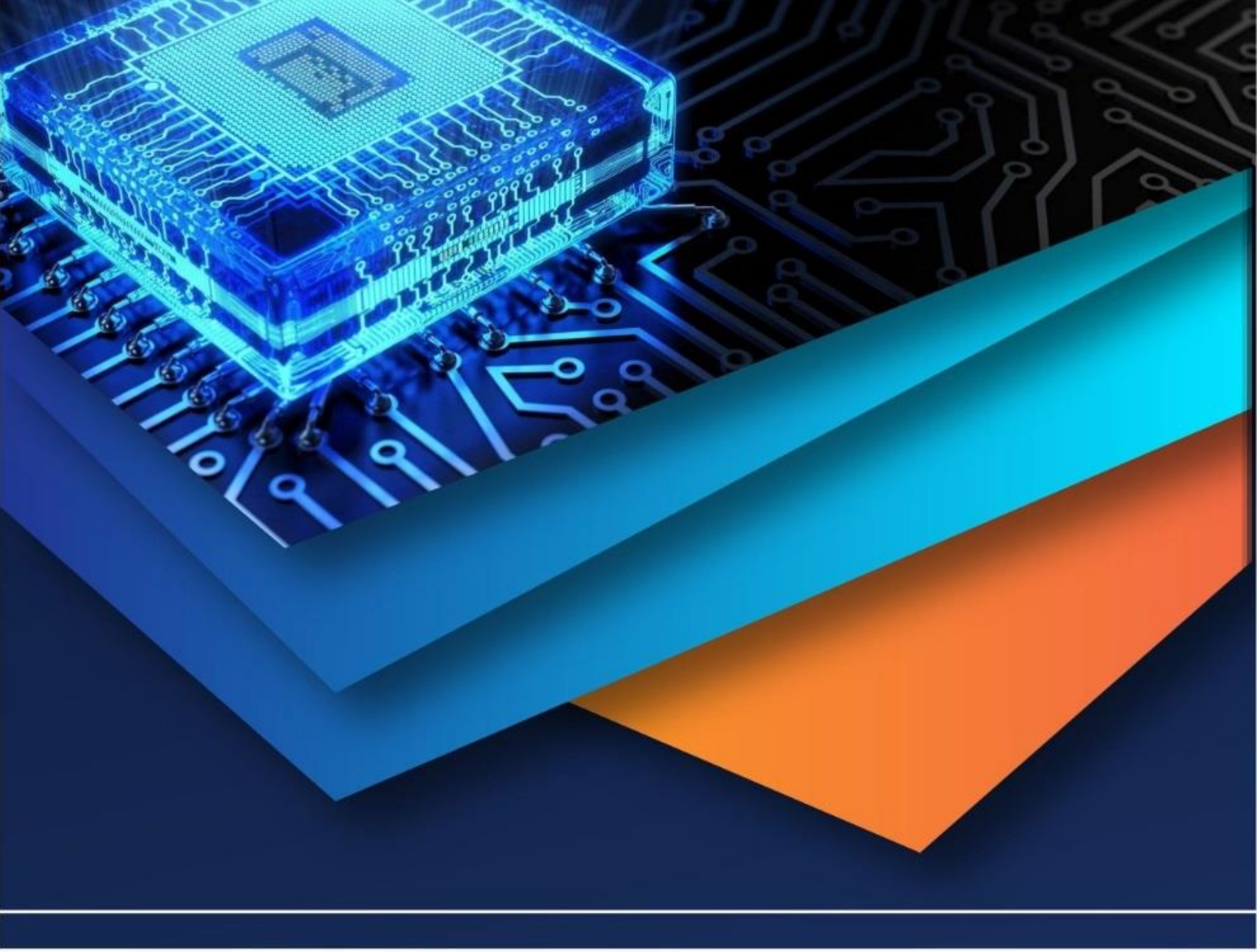
[4] Marín, R., Martínez, L., & Sanz, P. J. (2015). Design and implementation of an autonomous mobile robot navigation system using ROS. Journal of Intelligent & Robotic Systems, 80(2), 331-349.

[5] Pfeiffer, M., Beetz, M., & Dietrich, A. (2017). Scalable detection of objects in 2D laser range data for mobile robot navigation using ROS. Journal of Intelligent & Robotic Systems, 85(3-4), 571-591.

[6] Queralta, J. P., Benitez, A., Iborra, A., & Ollero, A. (2019). Sensor fusion-based reactive navigation system for outdoor autonomous robots. Robotics and Autonomous Systems, 120, 103259.

[7] Liu, H., Wang, D., Li, W., & Cao, Y. (2021). Autonomous navigation of mobile robot based on ROS and fuzzy control. Robotics and Autonomous Systems, 143, 103749.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)