www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

MULTIPLE MOTION CONTROL SYSTEM OF A ROBOTIC CAR BASED ON IOT TO PRODUCE CLOUD SERVICES

Bharath Kumar.P[1], Raviteja.Ch[2], Sai Vivek.G[3], Sirisaiah.L[4]

[1,2,3]UG Scholar, Dept. of EEE, St. Martin's Engineering College, Secunderabad, Telangana, India-500100
[4]Assistant Professor, Dept. of EEE, St. Martin's Engineering College, Secunderabad, Telangana, India-500100
lsirisaiaheee@smec.ac.in

Abstract*:*

IoT is a trending topic in todays world and it helps us in performing numerous task by making them easier and efficient. This paper proposes a robotic car where we will control its movement remotely using arduino uno controller. Client can manages the activities of the car from remote or far away spots over the remote correspondence using zigbee module and cloud service.The prime duty of this undertaking is that it utilize the viability of robot's development controlling structure in light of the fact that robotic auto can get immediate summons at once from fundamental sourceswhich make the moving framework more proficient. Commands and data are stored in local host service which delivers them when the device is ready to receive. We display the engineering and outline of the Arduino and wireless iot communication software and represent how to control the car by methods for commands and application.

Key Words: IoT,Robotic Car

1.INTRODUCTION

Nowdays use of robots which can be used remotely is getting popular especially in those fields where work posses hazard to human life. In such scenarios these robots can be used to perform work and replace human workforce and do work effectively .This venture proposes a numerous movement controlling component of a robotic auto utilizing Arduino UNO small scale controller.. Each device is uniquely identifiable by the controlling software which is the core concept of IoT. Client manages the activities of the car from remote or distant places over the wireless communication. There is boom in todays world for use of remotely operated robots which can do task more effectively and efficiently. These can be used to measure various variables from our environment and work on them. IoT is a very interesting and innovative concept which holds great future opportunities.

The main aim of this project is that it leverages the efficiency of robot's motion controlling system because robotic car can receive direct commands at a time from main sources which make the maneuvering system more efficient. Arduino is designed as an open-source electronics prototyping platform providing schematics and flexible development kits for enthusiastic users who intend to produce interactive objects or environments. Arduino can be used to sense surroundings by utilising various transducers to read and interpret inputs in order to make responses for example through the controlling of motors or transferring of data. In

today's world there is a significant development in the field of robotic control. Mobile robotic vehicles are light, small and

portable enough to be carried by an individual[5]. Our design serves as a solution to demonstrate how the control of the dc geared motors in coordination of the signals obtained from Wi-Fi module in conjunction of Arduino is used to achieve high degree of precise path control from the user side to achieve standard operations like moving at a particular target location, collecting data and avoiding any obstacle to prevent collision .

In existing literature many works have been done on the implementation and analysis of the robotics for various aspects like disaster management, working in nuclear areas, photography and military application. Cloud robotics uses computing resources to enable great memory, computational power and collective learning for robotics applications. When computational or storage demands exceed the on-board capacity of a robot, they are offloaded to the cloud, here the massive resources of a data centre can supplement the limited local resources of robots .Cloud robotics also represents a significant advance for robot learning[3]. The collision and detection protocol has been incorporated and well executed by using infrared sensors which prevents collision and sends the signal to the user mobile of obstacle detection. The robot direction can be changed by using the buttons available in the application .The robot is equipped with a pick and drop arm which can pick and drop up to 60 grams of weight. Our user end equipment mobile is equipped with an application control the path of the robotic car achieve its target location avoiding obstacles**.**

2. LITERATURE SURVEY

A smartphone is used to simplify and to increase the efficiency of processing of the voice commands. With their own independent OS and internet connection they are increasingly being used in numerous applications

One of the major features that we shall be making use of is - the Internet. An internet connection will allow the phone to communicate with the robot. Several Operating Systems are used for smart phones but the most common and efficient one is the Android OS developed by Google Inc. The Internet exchanges data easily and is a very proficient way of communication between two devices such as microcontroller and a smart phone

The robot can either maintain preset linear speed or can have variable speed onflat surfaces. The voice recognition is done with the help of a micro controller: NodeMcu. For detection and avoiding obstacles, an ultra-sonic module is implemented, which is programmed to stop the robot if there is any obstruction in its way, and it will inform the user to use another voice command[3]

Intelligent Flight Software Robot Based on Internet of Things by Yibin Hou , Jin Wang This paper proposed you can utilize matlab reenactment robot and programming, the utilization of cutting edge programming dialect to create clever flight robots. The examination aftereffects of insightful flight checking programming robot innovation can be connected in the scholarly world and industry and therapeutic transportation. This paper thinks about the clever flight observing programming robot under the Internet of Things

IOT Based Remote Access Human Control Robot Using MEMS Sensor by Abhishek Deendayal Patil, Husban Imtiyaz Kadiri, Ajinkya Shriram Joshi This paper proposed four different signals to operate the robot, forward, reverse, left and right. These commands are given by the user using the MEMS sensor. The MEMS Sensor will be set to the hand. At whatever point the hand moves toward some path, the mechanical development of the hand will be perceived by MEMS. MEMS interpret this mechanical hand development intoproportional electrical flags and send it to the Raspberry Pi. The Raspberry Pi at the transmitter side sends control signs to the recipient side through IOT (Internet of Things). The controller (ARM7) at the beneficiary zone gets these signs and provides guidance to the robot through IOT i.e. through cloud. Design and Construction oft Microcontroller Based Wireless Remote Controlled Industrial Electrical Appliances Using ZigBee Technology by Lu Mai , Min Zaw Oo This paper proposed microcontroller based remote controlled for electric frameworks parameters like voltage and current utilizing ZigBee innovation. PIC16F877A controller is utilized as a part of an overwhelming way since it is rich in peripherals and henceforth numerous gadgets can be interfaced quiet, it is likewise exceptionally shabby and can be effectively gathered and customized. The PIC controller controls the gadgets and sends the sensor esteems to the PC by means of ZigBee module. In spite of the fact that Bluetooth is superior to anything ZigBee for transmission rate, ZigBee has bring down power utilization.

3. PROPOSED METHODOLOGY

Phase 1: Requirements Gathering and System Design

1.1 Define System Objectives

- **Primary Objective**: Develop a robotic car that can be remotely controlled and monitored via a cloud platform.
- **Key Features**:
    - Remote control over the internet (via IoT).
    - Integration of multiple motion control techniques (differential drive, omni-directional drive).
    - Real-time data collection (location, speed, obstacles, etc.).
    - Cloud-based analytics and fleet management.

1.2 Functional Requirements

- **Motion Control**: Implement PID controllers for accurate motor control and trajectory planning algorithms for autonomous navigation.
- **Sensor Integration**: GPS for location, ultrasonic or LIDAR for obstacle detection, and IMU (Inertial Measurement Unit) for orientation and motion sensing.
- **Communication**: Real-time communication between the robotic car and the cloud using MQTT or HTTP protocols.
- **User Interface**: Web-based dashboard or mobile app for remote control and monitoring.

1.3 Hardware Design

- Select sensors (GPS, ultrasonic, IMU, camera).
- Choose appropriate microcontrollers (e.g., Raspberry Pi, Arduino, ESP32).
- Design motor control circuit for driving motors and actuators (e.g., servo motors for steering).
- Integration of wireless communication module (Wi-Fi, 4G/LTE).

Phase 2: System Architecture and Communication Protocols

2.1 Define IoT System Architecture

- **Robotic Car (Edge Device)**:
    - The car's microcontroller collects data from the sensors and controls actuators.
    - The IoT communication module (e.g., ESP32) connects the robotic car to the cloud via Wi-Fi, 4G, or LTE.
    - Local data processing is done using edge computing techniques to reduce latency for real-time control.
- **Cloud Infrastructure**:
    - Cloud platform (e.g., AWS IoT, Google Cloud IoT, Microsoft Azure IoT) for managing communication, storing sensor data, and providing analytics.
    - The cloud performs complex tasks, like data analysis, path planning, and controlling multiple vehicles in a fleet.
- **User Interface**:
    - A mobile app or web-based dashboard interacts with the cloud server, allowing

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

users to send commands (start, stop, change speed, set waypoints, etc.).
    - The interface allows for monitoring the vehicle's health, location, speed, and sensor readings in real-time.

2.2 Define Communication Protocols

- **MQTT**: Lightweight messaging protocol ideal for IoT. The robotic car can publish sensor data (like location and speed) to the cloud using MQTT. The cloud can also send control messages back to the car using MQTT.
- **HTTP/HTTPS**: For transmitting more extensive data (e.g., system logs, command updates, etc.) between the cloud server and user interface.

Phase 3: Motion Control and Path Planning

3.1 Motion Control Algorithm Design

- **PID Control**: Implement a PID control algorithm for precise control of motors. The PID controller will adjust the motor's speed based on the error between desired and actual speed/position.
- **Multiple Motion Control Strategies**:
    - **Differential Drive**: For a basic two-wheel system, control the speed of left and right wheels independently for steering and movement.
    - **Omni-Directional Drive**: Implementing an algorithm that allows for movement in any direction, using multiple motors or wheels.

3.2 Path Planning

- **Trajectory Planning**: Implement algorithms like **A*** or **Dijkstra's** to calculate optimal paths. The system can avoid obstacles and follow a set of waypoints defined by the user or autonomously planned.
- **Obstacle Avoidance**: Use data from ultrasonic or LIDAR sensors to detect obstacles and make real-time adjustments to the vehicle's path.

3.3 Sensor Fusion

- **IMU Integration**: Combine data from the accelerometer, gyroscope, and compass to provide a more accurate understanding of the car's orientation and movement.
- **GPS + Sensor Fusion**: Use GPS for coarse location data, but refine this with sensor fusion (e.g., IMU data) for more accurate position tracking.

Phase 4: IoT and Cloud Integration

4.1 Data Collection and Transmission

- The robotic car continuously collects data (location, speed, sensor readings) from onboard sensors.
- Data is transmitted to the cloud in real-time using IoT protocols (e.g., MQTT, HTTP).
- Data storage is organized on the cloud for easy access and analysis. For instance, Firebase or SQL databases can store the car's sensor logs, performance data, and historical movements.

4.2 Cloud-Based Control

- The cloud platform processes commands sent from the user interface (e.g., mobile or web app) and communicates control signals back to the car.
- Real-time control data is processed by the cloud for coordinated motion and optimal performance.
- The cloud also monitors the system's performance and alerts the user of any potential issues (e.g., low battery, maintenance needed).

4.3 Real-Time Monitoring and Analytics

- **Real-time Dashboard**: A user interface on a mobile app or web portal will display real-time data such as:
    - Current location on a map (using GPS).
    - Car's speed, battery level, and sensor readings.
    - System alerts (e.g., obstacle detected, low battery).
- **Analytics**: The cloud system will perform data analytics to provide insights on:
    - Performance trends (e.g., battery usage, travel efficiency).
    - Predictive maintenance alerts (based on usage data or abnormal sensor readings).

Phase 5: User Interface Design

5.1 Web/Mobile Dashboard

- **User Login**: Secure user authentication (OAuth, JWT) to ensure that only authorized individuals can control the robotic car.
- **Real-time Control**: The dashboard will allow users to:
    - Send control commands (move, stop, change direction).
    - View sensor data and system status in real-time.
    - Set waypoints or autonomous routes.
- **History/Log**: A log of historical movements and events is maintained for the user to review, including sensor data, vehicle health, and user commands.

Phase 6: Testing, Deployment, and Maintenance

6.1 Testing

- **Unit Testing**: Test individual modules like the motor control system, sensor integration, and cloud communication separately.
- **Integration Testing**: Test the complete system, including sensor data transmission to the cloud, real-time control commands, and data logging.
- **Performance Testing**: Assess the real-time responsiveness of the system. Ensure that the control signals and data transmission between the robotic car, the cloud, and the user interface occur with minimal latency.

6.2 Deployment

- Deploy the robotic car and the cloud infrastructure.
- Ensure smooth connectivity between the car and the cloud. Also, ensure the mobile app or web dashboard is functioning well.

6.3 Maintenance

- **OTA Updates**: Implement Over-the-Air (OTA) updates for firmware and software. This will allow the robotic car to be updated remotely.
- **Continuous Monitoring**: Use cloud analytics to monitor system health, track the car's location, and alert the user of any issues.

4. EXPERIMENTAL ANALYSIS

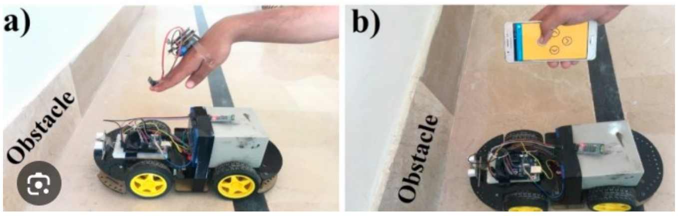
- **Experimental Analysis for the Multiple Motion Control System of a Robotic Car Based on IoT for Cloud Services :**Experimental analysis is crucial for assessing the performance, reliability, and accuracy of the robotic car system. It evaluates how well the robotic car performs its tasks in real-world scenarios and tests the effectiveness of each component in the system. The following experimental analysis would be performed at various stages of system development and deployment to validate the functionality and efficiency of the IoT-based motion control system integrated with cloud services.

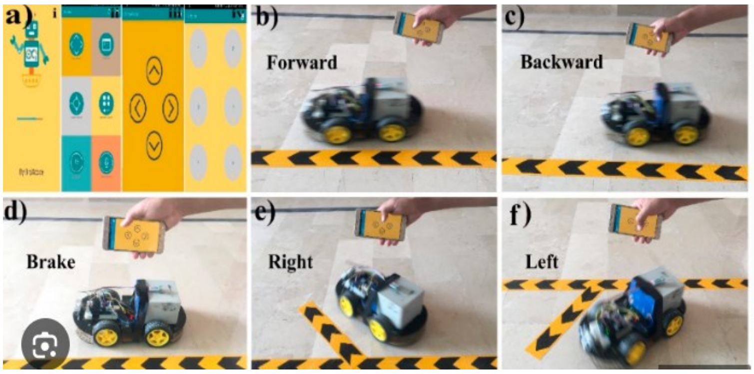
1. Experimental Setup and Parameters

- **Robotic Car Hardware:**
    - Sensors: GPS module, IMU (accelerometer, gyroscope), ultrasonic/LIDAR for obstacle detection.
    - Motor Controllers: For controlling the car's movement (differential drive or omni-directional system).
    - Microcontroller: Raspberry Pi, Arduino, or ESP32 for controlling the motors, reading sensor data, and handling communication.
    - Communication Module: Wi-Fi (ESP32) or 4G LTE module for communication with the cloud.

2. Performance Evaluation Criteria

- **Path Following Accuracy**: The robotic car is tasked with following a predefined path or trajectory, either autonomously or under remote control. The experimental test would compare the actual path of the car to the ideal path.
    - **Method**: A path-following algorithm (e.g., A*) will be tested to check how well the car follows the set trajectory under various conditions (e.g., obstacles, turns).





3. Data Collection and Analysis

- GPS Accuracy: Evaluate the positioning accuracy of the GPS module by comparing the car's reported location with the actual position (tracked manually or using an external system).
- Method: Use a GPS logger to capture both the real and expected GPS coordinates as the car moves along a path**.**

4. Cloud-Based Control and Monitoring Evaluation

4.1 Real-Time Monitoring and Control:

- UI Response Time: Test the real-time update frequency on the user interface (UI). This includes the rate at which the car's sensor data (e.g., location, speed) is updated in the dashboard and the delay between sending a command and the car executing it.
    - Method: Evaluate UI responsiveness with different network conditions (e.g., high latency, low bandwidth).

4.2 Fleet Management (If Applicable)

- **Multiple Vehicle Coordination**: In case of fleet management, test how well the cloud platform manages multiple robotic cars, sending coordinated commands and ensuring each car operates efficiently.
    - **Method**: Simulate a fleet of cars following different paths, avoiding obstacles, and synchronizing with cloud commands

5. CONCLUSION

The proposed robot can be used in war field, mines, power station, military operations, industries, research and educational institutions and so on. And also be used wherever people cannot go or where things doing too dangerous for humans to do safely. The Robotic movement is controlled remotely through the local system. It can do task more effectively and efficiently than human labor and also at same time take care of safety.

REFERENCES

[1] G. Amato, M. Broxvall, S. Chessa, M .Dragone, C.Gennaro, & C. Vairo, "When wireless sensor networks meet robots", The 7th International conference on Systems& Networks Communication (ICSNC2012), pp35-40.

[2] T. Braunl, Embedded Robotics; mobile robot design and applications with embedded systems", 2nd Edition, Springer, Germany 2006.

[3] LaMarca, W. Brunette, D. Koizumi, M. Lease, S.B. Sigurdsson, K. Sikorski, D. Fox, and G. Borriello, "Plant care: an investigation in practical ubiquitous systems", The 4th Intr. Conf. on Ubiquitous Computing, UK 2002, pp316-332.

[4] D. Floreano, F. Mondada, “Automatic Creation of an Autonomous Agent: Genetic Evolution of a NeuralNetwork Driven Robot”, SAB94 Proceeding of the 3rd Intr. Conference on Simulation of adaptive behavior, USA 1994,pp.421-430.

[5] R.C. Luo, and T.M. Chen, "Remote supervisory control of a sensor based mobile robot via Internet", Intr. Conf. on Intelligent Robots and Systems (IROS97), 1997

[6] M. Rahimi, H. Shah, G. Sukhatme, J.Heideman, and D.Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network", IEEE Intr. Conf. on Robotics &Automation (ICRA03), September 2003, pp.19-24.

[7] N. Wanga, N. Zhangb, and M. Wangc, "Wireless sensors in agriculture and food industry-Recent development and future perspective", Computers and Electronics in Agriculture , Vol.50, No.1, January 2006, pp1–14

[8] R.C. Luo, and T.M. Chen, " Remote supervisory control of a sensor based mobile robot via Internet", Intr. Conf. on Intelligent Robots and Systems (IROS97), 1997. X. Xue, S.X. Yang, M.Q.-H Meng, "Remote Sensing and Tele operation of A Mobile Robot Via the Internet", IEEE Intr Conf. on Information Acquisition, July 2005.

[9] G.M. Acaccia, R.C. Michelini, R.M. Molfino, and R.P. Razzoli, "Mobile robots in greenhouse cultivation: inspection and treatment of plants”, First Intr. Workshop on Advances in Service Robotics, 13-15.

[10] N. Yamasaki and Y. Anzai, “Active interface www.Howstuffworks.com for understanding microphone concepts, rf Dougherty, Kevin “Remote Control Robots Get Trial Run In Bosnia,” Stars and Stripes, 6 February 2000 March 2003, Italy.