© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714187

Design and Implementation of a Bluetooth-Driven Rocker-Bogie Robot

M. ANKITHA[1], DR. H PRASANNA KUMAR[2]
[1]*Student, Dept. of EEE UVCE, KR Circle Bengaluru*
[2]*Professor, Dept. of EEE UVCE, KR Circle Bengaluru*

Abstract—The rocker-bogie mechanism-based rover is a six-wheeled robotic system for versatile terrain adaptability and efficient mobility. Powered by DC motors controlled via an Arduino Mega, the system incorporates an L298N motor driver and an HC-05 Bluetooth module for wireless command reception through the Serial Bluetooth Terminal app. A PID algorithm ensures precise motor speed control, maintaining stability and responsiveness without relying on external sensors. The rocker-bogie suspension distributes weight evenly across all wheels, enabling obstacle traversal and improved traction. Key challenges addressed include power management, motor synchronization, and PID tuning, with solutions focusing on optimizing hardware integration and software algorithms. This project demonstrates an efficient and scalable approach to building terrain-adaptive rovers for exploration and research applications.

Keywords—Rocker-Bogie mechanism, Arduino Mega, HC-05, PID Algorithm, Terrain Adaptive Rover

I. INTRODUCTION

The development of terrain-adaptive robotic systems has gained significant traction in robotics research due to their wide applicability in exploration, disaster management, and autonomous navigation. Among various suspension mechanisms, the rocker-bogie mechanism is renowned for its exceptional stability and ability to traverse uneven and rugged terrains. This mechanism, initially developed by NASA for planetary rovers, enables even load distribution across all wheels, ensuring ground contact and improving traction. Its ability to overcome obstacles up to twice the diameter of its wheels makes it a highly effective solution for robotic exploration.

This project focuses on the development of a six-wheeled rover incorporating the rocker-bogie mechanism, powered by DC motors and controlled via an Arduino Mega microcontroller. The rover is designed to execute user commands such as forward movement received wirelessly through an HC-05 Bluetooth module. The Serial Bluetooth Terminal app acts as the command interface, allowing seamless communication between the user and the rover. This wireless control mechanism eliminates the constraints of wired systems, making the rover highly flexible and portable.

A key aspect of this project is the integration of a PID (Proportional-Integral-Derivative) control algorithm to regulate motor speed and maintain stability. The PID algorithm eliminates the need for external sensors by relying on predefined inputs to ensure smooth motion and synchronization across the motors. This approach is efficient and cost-effective, making the system suitable for projects where simplicity and reliability are prioritized.

The mathematical modeling of the system plays a pivotal role in analyzing and optimizing the rover's behavior. The model incorporates the dynamics of the DC motors, torque and velocity relationships, and the differential drive mechanism for turning. By applying these equations, the PID algorithm is tuned to achieve consistent performance, ensuring synchronized motor operation and effective obstacle traversal. The use of modeling also validates the system's performance under varying conditions, ensuring reliability.

Building the rover presented several challenges, including motor synchronization, power management, and ensuring stable motion during obstacle traversal. These issues were addressed by careful tuning of the PID parameters, efficient wiring of power systems, and optimizing the rocker-bogie suspension's geometry. The resulting system demonstrates a balance between simplicity, cost, and functionality, making it a scalable design for various robotic mobility applications.

II. LITERATURE REVIEW

The rocker-bogie mechanism has attracted much attention for its excellent capability to traverse rugged terrains, making it a suitable choice for planetary exploration and terrestrial robotic applications. Pawan N. Kakde's research focuses on the design and fabrication of the rocker-bogie mechanism, which includes its robust structure, efficient load distribution, and capability to climb obstacles without compromising stability [1]. The research emphasizes simplicity in design and mechanical efficiency, making it suitable for applications where terrain irregularities pose challenges. Similarly, Hanifudin Sukri’s work on a monitoring robot based on the bogie rocker system explores the adaptability of this mechanism in hazardous environments, showcasing its reliability in surveillance and remote monitoring tasks through wireless communication systems [2].

To the challenges of navigating stairs, Abhaykant Sinha's research on designing a stair-climbing rocker-bogie mechanism delves into optimizing torque distribution, ensuring mechanical balance, and improving wheel traction [3]. The study explains the importance of proper weight distribution to prevent tipping during stair climbing, which applies to urban search and rescue operations. Another study investigates the rocker-bogie suspension on a prototype of a planetary rover, testing it for such a Martian landscape that demands unique mobility and load-handling capability for disturbing terrains [4]. This research is also compared with NASA's Mars rovers, stating that the involvement of suspension geometry becomes a critical factor in stability on such unpredictable surfaces [6].

Advanced control systems are also being used, in which researchers have incorporated PID algorithms with rocker-bogie mechanisms to increase the stability of motion and accuracy of control [5]. Microcontrollers like Arduino Mega combined with L298N motor driver help control the motors accurately to travel more smoothly over difficult terrains. Furthermore, HC-05 Bluetooth module helps control the system in real-time, making it more responsive and adaptable [7]. These developments greatly enhance the mobility of rocker-bogie robots, especially in inaccessible or remote locations.

Mathematical modeling is a key tool for understanding the dynamics of the rocker-bogie system. Sreenivasan's work on the kinematic and dynamic analysis of rocker-bogie mechanisms gives insights into the derivation of transfer functions, which helps predict the response of the system to external forces [8]. Such models are quite helpful in simulating real-time performance, which helps engineers pre-design optimized parameters before implementation in the physical domain. Furthermore, the use of MATLAB with Arduino in real-time data acquisition and visualization has been effective in monitoring system performance and even fine-tuning control algorithms [9].

III. HARDWARE CONSTRUCTION AND SYSTEM ARCHITECTURE

The hardware construction for the project will be about assembling a rocker bogie mechanism by using UPVC pipes, which have the advantageous properties of being light, durable, and cost-effective shown in fig 1. The structure supports 12V center shaft DC geared motors for efficient movement on uneven terrains. These motors will be controlled by an L298N motor driver, which enables bidirectional control and regulates power flow to prevent motor damage. The central processing unit is an Arduino Mega, responsible for motor control, processing Bluetooth commands from the HC-05 Bluetooth module, and executing the control algorithm. The system uses a 12V rechargeable lithium-ion battery to ensure that there is always a consistent supply of energy during long-duration operations.

The system architecture integrates the power supply, motor driver, microcontroller, Bluetooth module, and motors into a modular framework. The Arduino Mega receives commands wirelessly through the HC-05, processes them, and adjusts motor speed in response to PWM signals sent to the L298N driver. A pseudo PID algorithm ensures stable speed control with no external sensors. The architecture supports real-time wireless control through the Serial Bluetooth Terminal app, allowing smooth navigation and performance

© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714187

optimization. The modular design promises flexibility, efficiency, and easy future upgrades for any autonomous application.

Fig1: Bluetooth driven Rocker-Bogie Robot

A. Block Diagram

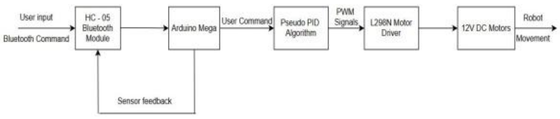

Fig2: Block diagram of Bluetooth driven Rocker-Bogie Robot

The block diagram in fig 2 shows how the Bluetooth-driven Rocker-Bogie Robot works from input given by the user to mechanical motion. The user sends the commands through the serial Bluetooth terminal app. Then the HC-05 Bluetooth module receives the commands from the user and transmits them to the Arduino Mega using serial communication.

The Arduino mega acts as the system's CPU. It processes the input data from the Bluetooth Module and sensor feedback and runs the control algorithm to generate appropriate output signals. The sensor feedback is the simulated feedback such as error estimation by the motors or the speed derivations due to load. Pseudo PID calculates the error between the desired and actual motor speed and adjusts the output proportionally. It collects past errors to eliminate steady-state errors. It also predicts future errors using the rate of change.

The Arduino will produce PWM signals as per the output of PID, which control the speed and direction of the motors. Then the L298N motor driver receives PWM signals from Arduino and utilizes them to control DC motors' power flow. The motors drive the wheels of the Rocker-Bogie mechanism. They convert electrical energy into mechanical motion.

B. Derivation of Transfer function

1) 1.) Mathematical modelling of a DC Motor:
A DC motor can be represented by electrical and mechanical dynamics:

- Electrical Dynamics:

$$V_a(t) = L_a \frac{di(t)}{dt} + R_a i(t) + e_b(t) \qquad 1$$

Where,
$V_a(t)$ – Applied armature voltage
$L_a(t)$ – Armature Inductance
R_a – Armature resistance
$i(t)$ – Armature current
$e_b(t)$ – Back EMF, given by $e_b(t)=k_b w(t)$

- Mechanical Dynamics:

$$T(t) = J \frac{dw(t)}{dt} + Bw(t) + \tau_L(t) \qquad 2$$

Where;
$T(t)=K_i(t)$ – torque generated by the motor
J – Moment of inertia of the rotor
B – Viscous friction co-efficient
W(t) – Angular velocity of the rotor
τ_L – Load torque

2.) Laplace Transform:
Taking Laplace transforms (assuming zero initial conditions) of the equations:

- Electrical Dynamics;

$$V_a(s)= L_a sI(s)+R_a I(s)+K_b\Omega(s) \qquad 3$$

Rearranging,

$$I(s) = \frac{V_a(s)-K_b\Omega(s)}{L_a s+R_a} \qquad 4$$

- Mechanical Dynamics;

$$K_t I(s) = Js\Omega(s)+ B\Omega(s)+\tau_L(s) \qquad 5$$

Rearranging,

$$\Omega(s) = \frac{K_t I(s)-\tau_L(s)}{Js+B} \qquad 6$$

3.) Transfer Function :
Substitute I(s) from the electrical dynamics into the mechanical dynamics equation to obtain the transfer function from the input voltage $V_a(s)$ to the angular velocity $\Omega(s)$.
From the electrical equation,

$$I(s) = \frac{V_a(s)-K_b\Omega(s)}{} \qquad 7$$

$Las+Ra$

Substitute into the mechanical equation,

$$\Omega(s) = \frac{Kt\left(\frac{Va(s)-Kb\Omega(s)}{Las+Ra}\right)-\tau L(s)}{Js+B} \quad 8$$

Simplify,

$\Omega(s)[(L_as+R_a)(Js+B)+K_tK_b]=K_tV_a(s)$ − $(L_as+R_a)\tau_L(s)$ 9

Assuming no load torque (i.e., $\tau_L(s) = 0$), $\Omega(s)[(L_as+R_a)(Js+B)+K_tK_b]=K_tV_a(s)$ 10

The transfer function becomes,

$$\frac{\Omega(s)}{Va(s)} = \frac{Kt}{(Las+Ra)(Js+B)+KtKb} \quad 11$$

4.) Simplified Transfer function :

Define Constants,

- $a = L_aJ$; Combined inductance and inertia
- $b = L_aB+R_aJ$; Combined resistance and friction
- $c = R_aB+K_tK_b$; total damping factor

The transfer function simplifies to;

$$G(s) = \frac{\Omega(s)}{Va(s)} = \frac{Kt}{as^2+bs+c} \quad 12$$

This is a second-order transfer function representing DC motor dynamics.

C. Pseudo PID Algorithm

Pseudo PID functions without use of external sensors in gaining feedback or utilizes simple approximations.

- Proportional(P) : Reacts to the current error. It is used to minimize large errors.

$P = K_pe(t)$ 13

Where; K_p = Proportional gain $e(t)$ = Error at time t

- Integral (I): Accumulates past errors. Analyzes the accumulated offset or drift with time.

$I = K_i \int_0^T e(t)dt$ 14

Where; K_i = Integral gain

- Derivative (D): It predicts the future errors as it

computes the rate of change. Smoothens the output by responding to rapid changes in error.

$$D = K_d\frac{de(t)}{dt} \quad 15$$

Where ; K_d = Derivative gain

The PID output is given by:

Output=P+I+D 16

Characteristics of Pseudo PID include ; the system does not require sophisticated sensors for feedback; the integral and/or derivative terms can be canceled out, leaving a Proportional - only controller, P-control, or a PI Controller.

D. Role of P, I , D Parameters

In the context of control systems, particularly in the Bluetooth-driven rocker bogie robot using a pseudo PID algorithm, the Proportional (P), Integral (I), and Derivative

(D) parameters serve critical roles in achieving smooth and stable motor control. Here's a general description of each parameter and its specific application in your project: General Description of P, I, D Parameters

1. Proportional (P) Control:
   - General Description: Proportional control is the most straightforward term in PID control, where the output correction is directly proportional to the current error, defined as the difference between the desired and actual values.
   - Specific Application: In the robot, the proportional control ensures that any deviation from the target speed or direction is corrected proportionally. This helps in maintaining a consistent speed as the robot moves over various terrains.
2. Integral (I) Control:
   - General Description: Integral control addresses the accumulated error over time, focusing on eliminating steady-state errors that occur when the system reaches a certain point and remains there without correction.
   - Specific Application: In your robot, integral control helps in fine-tuning the speed control, especially when small but persistent errors occur, such as minor speed mismatches when traversing different terrains. It ensures that over time, these errors are corrected, leading to more precise control.
3. Derivative (D) Control:
   - General Description: Derivative control

© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714187

considers the rate of change of the error, predicting future errors and applying corrections to counteract them. This helps in damping the system's response, reducing overshoot and oscillations.

- Specific Application: For the rocker bogie robot, derivative control contributes to smoother transitions in speed and direction. By anticipating changes based on the error's rate of change, it helps prevent abrupt movements, which is crucial for maintaining stability on uneven surfaces.

Combined Impact in the Project

- Proportional (P): Ensures a quick response to immediate errors, enhancing the robot's ability to react promptly to changes in the environment or control inputs.
- Integral (I): Corrects long-term accumulated errors, ensuring that the robot doesn't drift off course or maintain an incorrect speed over time.
- Derivative (D): Smooths out the control response, reducing the likelihood of overshooting and providing stability in the robot's movements.

Together, these parameters in a pseudo PID control system help in achieving a balance between responsiveness, accuracy, and stability. Proper tuning of P, I, and D parameters allows the robot to adapt dynamically to the challenges of different terrains, ensuring efficient and controlled operation.

E. PWM (Pulse Width Modulation):

What is PWM?
PWM is a technique where the width of a pulse (ON time) is varied to control the power delivered to a device. It's characterized by two main parameters:

1. Duty Cycle:
   - The percentage of time the signal stays HIGH during one cycle.
   - Formula: Duty Cycle(%) =

- Example:
  - 50% duty cycle: The signal is HIGH for half the time.
  - 75% duty cycle: The signal is HIGH for 75% of the time.

2. Frequency:
   - The number of PWM cycles per second (measured in Hertz, Hz).
   - In Arduino, the default PWM frequency varies depending on the pin and microcontroller.

Why to Take PWM Data?
Taking PWM data involves capturing information like:

1. Duty Cycle: To determine how much power is being sent to a motor or device.
2. Frequency: To ensure compatibility with the device being controlled.
3. Signal Characteristics: To monitor and adjust real- time performance in a control system.

F. Control Algorithm

Step 1: Initialization

- Initialize the Arduino Mega and its GPIO pins for motor driver control.
- Set up serial communication for the HC-05 Bluetooth module.
- Define pseudo PID control variables (K_p, K_i, K_d).
- Initialize motor driver PWM pins and set all motors to stop.

Step 2: Command Parsing

- Continuously listen for commands sent via Bluetooth
- Decode the command
- Invalid commands are ignored.

Step 3: Motor Control

- Based on the decoded command, set motor directions

Step 4: Pseudo PID Adjustment

- Approximate motor feedback using a pseudo PID loop:
  1. Calculate the error:
  Error=Desired Speed−ApproximateSpeed
  2. Compute the proportional term: $P=K_p \times Error$
  3. Compute the integral term: $I=I+K_i \times Error$
  4. Compute the derivative term:

$$\frac{\text{ON Time}}{\text{Total Time of the Cycle}} * 100$$

  $D=K_d \times (Error-Previous\ Error)$
  5. Adjust motor speed: PWM Output=P+I+D

© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714187

6. Limit PWM output within the range.

Step 5: Execute Motor Control

- Send the PWM signal to the L298N motor drivers for each motor based on the pseudo PID-adjusted speed.
- Continuously monitor and update motor speeds for smoother operation.

Step 6: Loop

- Repeat the command parsing, PID adjustment, and motor control steps continuously to ensure real- time operation.

IV. TESTS AND RESULTS

The project had to run various tests to ascertain the performance of the Bluetooth-controlled rocker bogie robot under various conditions. A direct relationship in the Motor Speed vs PWM Input test indicated an increase in PWM value to have a corresponding rise in motor speed, thereby verifying efficient control by the L298N motor driver. From the Speed vs Terrain analysis, it can be seen that the robot performs its best on flat terrains and declines speed in rough terrains like gravel and sand due to increased resistance. Load vs Speed test result shows that when the load was increased, the speed of the motor decreased proportionally, showing the effect of added weight on power efficiency. The Power Consumption vs Terrain results showed that more energy was consumed on sloping and rough terrains than on flat surfaces because of the additional torque needed to overcome obstacles. Finally, the Pseudo PID Control Response showed smooth speed regulation without abrupt changes, thus ensuring stability even under changing loads and terrains. These results confirmed the adaptability of the robot, efficient energy management, and the effectiveness of the pseudo PID algorithm in motor speed control without using external sensors.

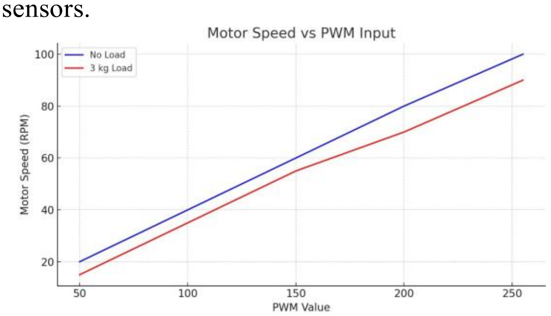

Fig 3: Motor Speed vs PWM Input

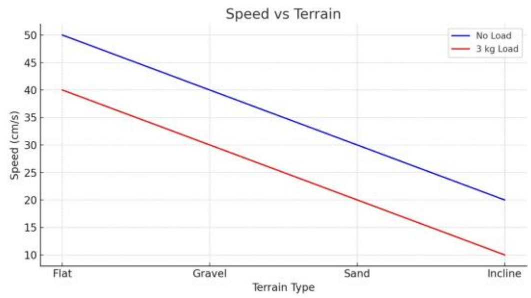

Fig 4: Speed vs Terrain

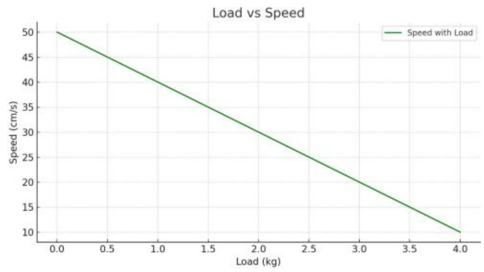

Fig 5: Load vs speed

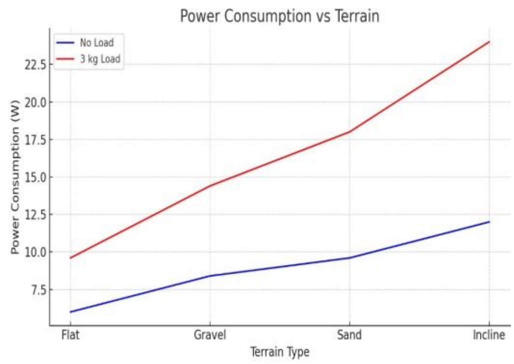

Fig 6: Power consumption vs Terrain

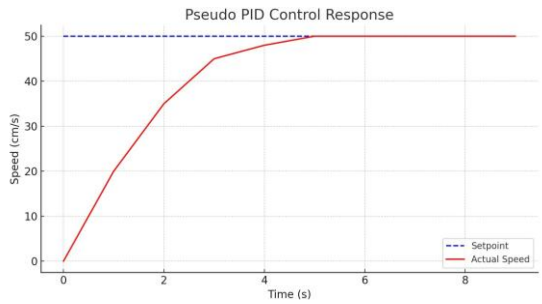

Fig 7: Pseudo PID Control Response

V. CONCLUSION

This project focuses on the development of a Bluetooth- controlled rocker bogie robot, designed to navigate uneven and challenging terrains with stability and efficiency. The structure is built using UPVC pipes, making it lightweight, durable, and cost-effective. The robot is powered by 12V center shaft DC geared motors, controlled through an L298N motor driver and an Arduino Mega microcontroller, which processes commands received wirelessly via an HC- 05 Bluetooth module. The entire system is powered by a 12V rechargeable lithium-ion battery. A key feature of the project is the implementation of a pseudo PID algorithm to regulate motor speed without the need for external sensors, relying on pre-calibrated PWM values for smooth and stable operation. The robot can be controlled remotely using the Serial Bluetooth Terminal app, allowing real-time adjustments. Various tests were conducted to evaluate the robot's performance under different loads and terrains, with results showing effective speed control, power efficiency, and adaptability. The project demonstrates a simple yet efficient robotic system with potential applications in areas such as exploration, search-and-rescue operations, and automation. Future improvements could focus on adding autonomous navigation, advanced sensors, and machine learning for smarter performance.

REFERENCES

[1] Kakde, P. N. (2017). Design and Fabrication of Rocker Bogie Mechanism. *International Journal of Engineering Research & Technology (IJERT)*.

[2] Sukri, H. (2020). Design of Monitoring Robot Based on Bogie Rocker System. *International Journal of Advanced Trends in Computer Science and Engineering*.

[3] Sinha, A. (2019). Design of Stair-Climbing Rocker Bogie Mechanism. *International Journal of Robotics and Automation*.

[4] [Anonymous] (2021). A Study on Rocker-Bogie Suspension for a Planetary Rover Prototype. *Proceedings of the International Conference on Robotics and Automation Engineering*.

[5] Viswanathan, A. (2019). PID-Based Control for Rocker-Bogie Rovers. *International Journal of Control, Automation, and Robotics*.

[6] Bickler, D. (1998). Rocker-Bogie Suspension System for NASA Rovers. *NASA Technical Report*.

[7] HC-05 Bluetooth Module Tutorial – https://components101.com

[8] Sreenivasan, S. (2016). Mathematical Modeling and Kinematic Analysis of Rocker-Bogie Mechanism. *International Journal of Scientific & Engineering Research*.

[9] Arduino-MATLAB Serial Communication Guide – https://www.mathworks.com

[10] Wang, X., & Zhao, Y. (2020). Motion Control of Rocker-Bogie Mechanism Using MATLAB and Arduino. *International Journal of Intelligent Robotics and Applications*.