IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, June 2026
Impact Factor: 8.2

Design and Development of Industrial Quadruped Robot

Apeksha Vilas Shelar[1], Sanket Vinayak Achpale[2], Yogita Deepak Kasabe[3],
Aditya Prabhashankar Jadhav[4], Prof. Abhishek Kumar Upadhyay[5]
[1,2,3,4] Student, Department of Automation & Robotics Engineering
[5]Professor, Department of Automation & Robotics Engineering
Pravara Rural Engineering College, Loni, India
[1] shelarapeksha611@gmail.com, [2] sanketachpale2005@gmail.com, [3] kasabeyogita87@gmail.com, [4] adityaprabhashankar@gmail.com, [5] upadhyayar@pravaraengg.org.in

Abstract: *In this paper, we present an industrial quadruped robot designed to navigate uneven and rough terrain. The primary objective of this system is to operate in environments that are difficult or unsafe for human access. The robot utilizes a Raspberry Pi 5 for high-level processing, a Teensy controller for motion control, and an onboard camera for monitoring. It is capable of walking, running, and jumping to adapt to various terrain conditions. Experimental results demonstrate that the proposed system achieves improved stability and performance compared to traditional wheeled robots.*

Keywords: Quadruped Robot, Gait Planning, Mobile Robotics, Raspberry Pi 5, Terrain Adaptation, Locomotion, Unstructured Terrain, Teensy.

I. INTRODUCTION

Robotic systems are increasingly deployed in environments that are hazardous or inaccessible to humans, such as disaster zones, industrial sites, and uneven outdoor terrains. Conventional mobile robots typically rely on wheeled locomotion, which is efficient on flat surfaces but performs poorly on irregular, rocky, or unstable ground. This limitation motivates the develop ment of alternative robotic platforms capable of operating in complex and unstructured environments. Quadruped robots, which utilize four-legged locomotion, provide a more stable and adaptable solution for such scenarios. Their design allows them to maintain balance, adjust posture, and negotiate obstacles more effectively than wheeled platforms. By leveraging multiple gait patterns—including walking, running, and jumping—they can traverse diverse terrains with greater robustness and reliability.

In this work, an industrial quadruped robot is developed using a combination of embedded systems and mechanical design. A Raspberry Pi is used for processing and decision-making, while a Teensy 4.0 microcontroller manages precise motor control. The system also includes sensors such as LiDAR and a camera to detect surroundings and assist in navigation. Servo motors and supporting mechanical components enable coordinated leg movements for stable locomotion. The aim of this project is to design a reliable robotic system that can move efficiently on unsurfaced terrains and perform tasks such as inspection, monitoring, and rescue. This approach improves mobility and extends the usability of robots in real-world environments where traditional systems are not effective.

II. LITERATURE REVIEW-

Before developing our system, we studied types of robotic locomotion systems to understand their advantages and limitations. many researchers working on different quadruped robot locomotion, gait planning, their stability and reinforceme nt learning. mostly traditional robots use cameras and visual sensors to detect the obstacles.The BOSTON DYNAMICS developed a product SPOT in 2019 an agile mobile robot for industrial inspection application (ICRA 2020). Demonstrated autonomous walking, stair climbing, obstacle avoidance, and industrial inspection using advanced

DOI: 10.48175/IJARSCT-37308

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, June 2026
Impact Factor: 8.2

sensors. The scientist M. Hutter et al. and his team worked on the blind quadruped movement using sensors instead of camera it shows it walks on rough areas successfully. the scientist J. Lee et al. used the reinforcement learning to improve robot walking. the robot learned through repeated practice, trial and error to achieve better stability and smooth movement. Many scientists and engineers have worked on making four-legged robots walk better on rough roads, stairs, and uneven surfaces. In the beginning, their main focus was on helping the robot keep balance while walking. They studied how the robot should place its legs so that it does not fall. This basic idea helped in improving robot movement.

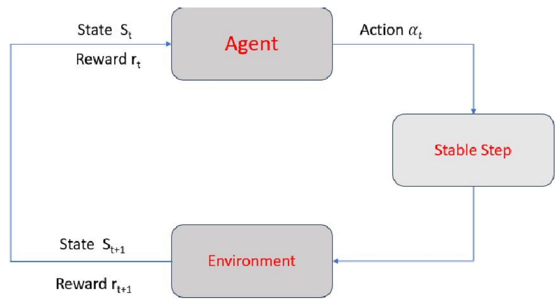

Fig 1. Block diagram of the reinforcement learning and stable step combination.

After that, researchers started testing robots in computer simulations before using them in real life. These simulations helped them understand how robots walk, climb, and react to obstacles. It also saved time and reduced mistakes during testing. Some robots use cameras to see and move, but cameras do not work well in dark places, smoke, dust, or disaster areas. Because of this, researchers developed something called blind gait. In blind gait, the robot does not depend on cameras. Instead, it uses sensors in its legs and body to feel the ground and maintain balance. This makes the robot stronger and more reliable in difficult environments.

Later, machine learning and reinforcement learning were used to make robots smarter. In this method, the robot learns by trying again and again, just like humans learn from practice. It checks which step is correct and improves its walking over time. This helps the robot move better on stairs and difficult paths. Some researchers combined both methods—stable blind gait and reinforcement learning. First, the robot gets a basic stable walking pattern, and then reinforcement learning helps improve it further. This gives better results than using only cameras or only learning methods. In this project, the same idea is used. The robot uses stable blind gait with reinforcement learning so that it can climb stairs safely and move on uneven surfaces without depending too much on cameras. This improves stability, safety, and performance in real industrial environments.

III. MATERIALS AND METHODS-

3.1. Kinematic Model-

The kinematic version is the basis for controlling the motion of the quadrupedal robot. It facilitates the knowledge of the relationship between the robot body, joints, and foot positions during locomotion. Each foot is composed of several joints th at movement. Many 2 shows the kinematic design of the robot, where the body frame and bone coordinate structure are described for accurate motion planning. The position of each foot tip is

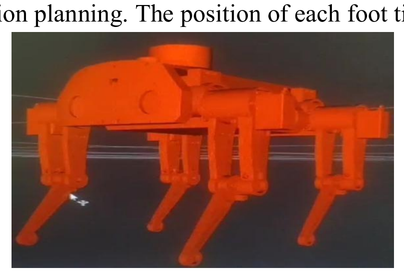

Fig 2. Quadruped Robot Kinematic Model

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, June 2026
Impact Factor: 8.2

calculated with the use of forward and backward kinematics, which allows the robot to maintain stability while adapting to hard surfaces. This version is especially critical for blind locomotion due to the fact that the robot cannot rely on visual sensors alone. Instead, it uses joint positions, force feedback, and posture to make movement choices. A robust kinematic framework improves stability, accuracy, and obstacle-control functionality.

3.2. Stable Blind Gait-

The static blind gait is a way that robots walk. Robots move without using cameras. The robot does not look at its surroundings. Instead, the static blind gait method uses. the robot's senses, like where its jointers what its legs are feeling, to know where the robot is and how to walk. The robot that uses the blind gait is always very steady. Before the robot moves one leg it makes sure that its other legs are holding it up firmly. This makes the robot walk slowly and carefully so the robot does not fall down. The static blind gait is good because the robot can stay balanced even if the robot stops moving. the static blind gait is very useful in some situations. For example, it is useful in the dark or in places with a lot of dust or on stairs or uneven ground. The robot learns to feel the ground and adjust its steps. The robot gets to know how to walk on types of ground.

The static blind gait is slower than ways that robots can walk. The static blind gait is more reliable and safer. People use the blind gait in situations where it is more important to be safe than to be fast, like when robots are used to inspect things in a factory or to help in emergencies or to navigate in places that the robot does not know. The static blind gait is a choice when safety is important.

3.3. Static Stability Analysis-

static stability analysis means checking whether robot stays balance without falling stays stable. In this robot Centre of mass(C0M) creates the polygon if the polygon changes its moves the polygon .it focuses on how the robot distributes its weight and how well its legs support the body at any given moment. The main idea behind this analysis is the concept of of support polygon. This polygon is formed by connecting the points where the robot's feet are in contact with the ground.

Table 1. Swing leg trajectory parameters-

Parameter	Explanation
dx	The displacement in the x-direction
dy	The displacement in the y-direction
h	The height of leg movement

The quadratic robot while walking carefully shifts its weight before lifting any leg. This shows the centre of mass (CoM) always stays stable and inside the loop maintaining continuously stability because of that robot even stop while coming obstacle without falling. The static analysis does not Consider any dynamic effects such as momentum, acceleration or external disturbances. Overall, the static analysis focus on stability. Simple way to design specially on the uneven terrain.

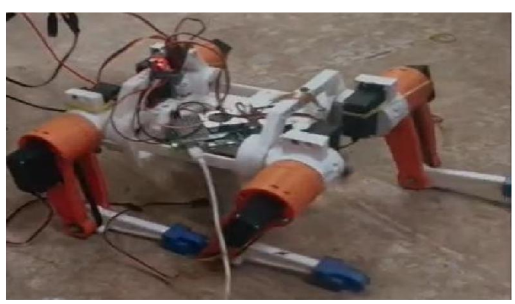

Fig 3. Stability Condition

3.4. Swing Leg Trajectory-

The swing Leg Trajectory describes how the robot moves it's one of the legs through the air while walking. When it lifts the leg from the ground it forms the planned polygon to stable the movement its centre of mass (CoM) moves and gives the stable movement. It follows the planned path before touching to the ground. The trajectory is created using the parabolic tit helps the leg moves naturally like lifting up, moving forward and then lifting down smoothly without any sudden movement.

Algorithm 1. Stable Step Algorithm

Require:
1: leg_id (Target leg identifier)
2: Δx, Δy (position adjustment)
3: h (Target foot height)
4: € (Ground force threshold) Ensure:
5: Stable leg placement while maintaining dynamic balance
6: Compute Center of Pressure (CoP) from support polygon
7: Adjust center of mass (CoM) to align with CoP: CoM
← CoP
8: while $Fz \leqslant \epsilon$ do
9: Compute inverse kinematics: θ ← IK (leg trajectory) 10: Apply joint positions: J ← θ
11: end while 12: if $Fz > \epsilon$ then
13: Confirm stable ground contact 14: return Success
15: else
16: return Failure (Unstable contact detected)
17: end if

Table 2. Symbols and parameters for Algorithm 1.

Symbol	Definition
leg_id	Identifier of the leg to be moved (1–4)
dx	Desired displacement in the x-direction (world frame
dy	Desired displacement in the y-direction (world frame)
h	Height of the leg movement during the swing phase
CoP	Center of Polygon, the ideal CoM position for stability
CoM	Center of mass of the robot
Fz	Vertical force sensor reading at the leg tip
ϵ	Force threshold for ground contact detection
θ	Vector of joint angles calculated by inverse kinematics
Ji	Joint angle for joint i (after IK calculation)

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 11, June 2026 Impact Factor: 8.2

IV. ALGORITHM STRUCTURE

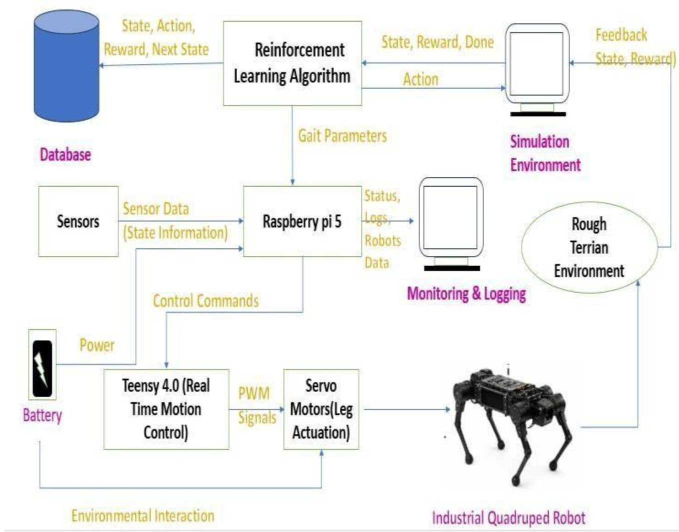

Fig4. working architecture of a Reinforcement learning based Industrial Quadruped Robot.

In the Fig 5.4 system uses reinforcement learning to train an industrial quadruped robot model on a rough terrain area. The robot learns to takes actions, gets feedback and improves its behavior over time. In the overall the sensors collect data and send to raspberry pi 5 then it processes data and sends to RL algorithm. It decides action. This action sent to simulation(training), Raspberry pi for real execution. Raspberry sends commands to teensy generates PWM. feedback(reward/state) sent back to RL. Learning improves continuously.

V. ALGORITHM PSEUDOCODE

Algorithm 3. Reinforcement learning with stable step

1: for each training episode do
2: Initialize simulation environment
3: Get initial state s
4: while robot has not fallen AND target not reached do
5: Sample action = [leg identifier, Δx, Δy, height] $\sim \pi(\bullet|s)$
6: Execute stable step sequence:
7: Compute Center of Polygon (CoP)
8: Align center of mass: CoM new $\leftarrow$ CoP
9: while ground reaction force Fz $\leqslant \epsilon$ do
10: Calculate joint angles: θ = IK (leg trajectory) 11: Update joint positions: Ji = θi
12: end while
13: Observe resulting states$'$
14: Calculate reward r for transition
15: Update policy π using experience tuple (s, action, r, s$'$)
16: Update current state: s $\leftarrow$ s$'$
17: end while
18: end for

Table 3. Symbols and parameters for Algorithm 2

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, June 2026
Impact Factor: 8.2

Symbol	Definition
episode	One complete training iteration
s	Current state observed by the agent
a	Action selected by the agent's policy
π(·|s)	Policy function, probability of actions given state s
r	Reward received after taking action a
s′	Next state observed after taking action a
leg_id, dx, dy, h	Components of action a, parameters for stable step (Algorithm 1)

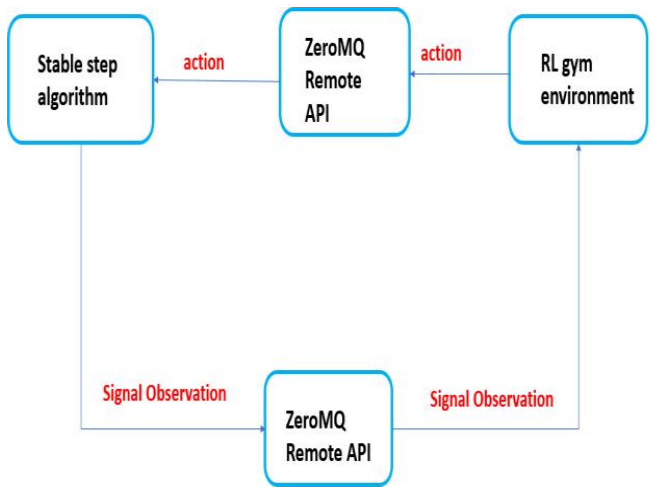

Fig5. System Architecture of Simulator and Python API Integration.

VI. RESULT

The staircase used in this study was ten centimeters high. The agent had to get to the top of the staircase. To compare we looked at how the robot moved when it used a walk, with fixed settings, which is shown in the extra videos. The agent wanted to get all four of its legs over the staircase. We used a system to track how the agent learned. At the end of each try we wrote down a zero or a one. If the agent got to the top of the staircase, we wrote down a one. If it did not make it, we wrote down a zero. This simple way of tracking helped us see how the agent learned to climb the staircase.

VII. CONCLUSIONS

This project showed a way to help a four-legged robot move around. It mixed two methods: using math to plan the robot's movements and using machine learning to help the robot learn. The robot can walk steadily on ground because of a special walking method. This method uses math to control the robots' legs, balance and movement. The project also taught the robot to climb stairs using machine learning. This helped the robot make decisions in tricky situations. The robot was able to climb stairs 93% of the time. This work is important because it shows that using both math and machine learning can help robots move around well. This can be used in real-life jobs like checking buildings finding people and exploring areas. It can also help make robots that can move around on their own. The four-legged robot and stair climbing were key to this project. The robot's ability to walk and climb was tested times. The results showed that the robot can move around well on ground and climb stairs. The project used a four- robot for testing. The robots' movements and stair climbing were tested times.

This project made a progress, in robotics. The four-legged robot and machine learning were used together. The results were good. Showed that this method can help robots move around well. The project can be used for real-life applications. It can help robots do tasks on their own. The project showed that a four-legged robot can move around well.

It can walk on ground and climb stairs. The robots' movements were controlled using math and machine learning. The results were good. Showed that this method is effective. The project can help make robots that can move around on their own. It can also help robots do tasks. The four-legged robot was the part of this project. The robots' movements and stair climbing were tested times. The results showed that the robot can move around well. The project used machine learning to help the robot learn. The results were good. Showed that this method can help robots move around well.

VIII. ACKNOWLEDGMENT

We would like to express our sincere gratitude to our project guide, Prof. Abhishek Kumar Upadhyay, for their continuous support, valuable guidance, and encouragement throughout the development of this project. Their insights and suggestions greatly contributed to the successful completion of this work. We are also thankful to the Department of Automation and Robotics Engineering for providing the necessary resources and support required for this project. We extend our gratitude to all facul ty members and our friends for their cooperation and assistance.

REFERENCES

[1]. K. Irie et al., "Rough terrain navigation for a quadruped robot using deep reinforcement learning," Advanced Robotics, 2025. [2]. Z. Ke and H. Ma, "Reinforcement learning based path planning for quadruped robot," Springer, 2025

[3]. L. Zhao, Y. Lin, "Typical failure analysis and processing of belt conveyor," Procedia Engineering, vol. 26, pp.942–946, 2011, Doi: HTTPs://doi.org/10.1016/j.proeng.2011.11.2260.

[4]. M. R. Sayali Todkar, "Design of Belt Conveyor System," International Journal of Science, Engineering and Technology Research, vol. 7, no. 7, pp.458–462, 2018.

[5]. P. Ela Murugan et al., "Automatic material segregation using PLC," International Journal of Engineering and Technology (UAE), vol. 7, no.2,pp.376380,2018,doi:https://doi.org/10.14419/ijet.v7i2.24.12088.

[6]. PSG college of engineering, 307Design_Data_Data_Book_Of_Engineers_By_PSG (Coimbatore, India: Kalanithi Atchara).

[7]. K. S.J. Ojolo, J.I. Rosalee, Adelaja, A.O., "Design and Development of Waste Sorting Machine," Journal of Emerging Trends in Engineering and Applied Sciences (JETEAS), vol. 2, no. 4, pp. 576–580, 2011.

[8] Bhandari V. B., Design of Machine Elements - V. B. Bhandari - Google Books (1994)

[8]. M. Salahuddin, "Adaptive motion planning for legged robots," Nature Scientific Reports, 2026.

[9]. U. Ranasinghe et al., "Review of reinforcement learning for quadruped robots," SSRN, 2025.

[10]. P. K. V. Kona Kalla Naga Sri Ananth, Vaila Rakesh, "Design and Selecting the Proper Conveyor-Belt," International Journal of Advanced Engineering Technology E, vol. IV, no. II, pp. 43–49, 2013.

[11]. J. Hwangbo et al., "Learning agile and dynamic motor skills," Science Robotics, 2019.

[12]. J. Tobin et al., "Domain randomization for transferring deep neural networks," IROS, 2017.

[13]. T. Haarnoja et al., "Soft Actor-Critic Algorithms," ICML, 2018

[14]. T. Haarnoja et al., "Soft Actor-Critic Algorithms," ICML, 2018

[15]. M. Salahuddin, "Adaptive motion planning for legged robots," Nature Scientific Reports, 2026.

[16]. Y. Zhang et al., "Dual-layer reinforcement learning for quadruped locomotion," MDPI, 2024.

[17]. Y. Yang and Z. Zhang, "Rough-terrain path planning using deep RL," MDPI, 2025.

[18]. J. Schulman et al., "Proximal Policy Optimization Algorithms," arXiv, 2017.

[19]. T. Lillicrap et al., "Continuous control with deep reinforcement learning," arXiv, 2015.