JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# Design and Development of an AI-Based Hexapod Robot with Robotic Arm for Autonomous Navigation and Object Manipulation

**1st Preeti Kumari**
*Dept. of ECE*
*Govt. Engg. College West-Champaran*
Bettiah, Bihar, India
preetycit2004@gmail.com

**2nd Jyoti Mala**
*Dept. of ECE*
*Govt. Engg. College West-Champaran*
Bettiah, Bihar, India
jyotiesselutility@gmail.com

**3rd Alok Kumar**
*Dept. of ECE*
*Govt. Engg. College West-Champaran*
Bettiah, Bihar, India
Sayes2aloksahu@gmail.com

**4th Keshab Kumar**
*Dept. of ECE*
*Govt. Engg. College West-Champaran*
Bettiah, Bihar, India
keshabchoudhary1992@gmail.com

**5th Saket Kumar**
*Dept. of ECE*
*Govt. Engg. College West-Champaran*
Bettiah, Bihar, India
sgsaket1@gmail.com

*Abstract*— **This paper presents the design and development of a low-cost, AI-based hexapod robot with an integrated robotic arm for autonomous navigation and object manipulation. Unlike wheeled or tracked robots, the six-legged hexapod provides improved stability on uneven, rocky, and soft terrain using a tripod gait, where three legs remain in contact with the ground during movement. A 4-DOF robotic arm mounted on the platform enables pick-and-place operations, combining mobility and manipulation in a single system. The robot uses a Raspberry Pi-based vision module with YOLOv8 for real-time object detection, identifying obstacles and target objects from camera input. A rule-based decision-making algorithm generates commands for navigation, obstacle avoidance, and object handling. These commands are sent to an Arduino Mega controller through UART, while NRF24L01 wireless communication supports remote control. The robot is built using lightweight 3D-printed PLA+ components and servo motors, with a total hardware cost below ₹20,000. Experimental results show stable locomotion, 85% mAP object detection, and reliable manipulation. The system is suitable for search-and-rescue, hazardous-environment exploration, surveillance, industrial automation, and agricultural applications.**

**Keywords— Hexapod Robot, Robotic Arm, YOLOv8, Autonomous Navigation, Embedded System, Object Detection**

## I. INTRODUCTION

Robotics has been recognized as an important technology in automation, capable of addressing key challenges across a range of applications. From rescue operations in disaster zones to industrial automation in manufacturing plants, robots are now used to enhance human capabilities in handling complex situations while limiting human involvement. In high-risk regions such as nuclear plants or volcanic regions, robots have been used to explore these areas with ease [1, 19]. The traditional robots used in these applications have been wheeled or tracked and have been effective only on smooth, flat surfaces. However, these robots are very effective at moving across terrain with obstacles such as rocks or soft ground. As a result, these robots have been found to fail in mobility, payload capacity, or efficiency. In contrast, legged robots have been effective in dealing with such rough terrain owing to their adaptability in changing their foot placement or posture. Among these robots, hexapod robots with six legs have been effective owing to their stability. The hexapod robots use the tripod gait algorithm in which three legs are in contact with the ground at any given time [9, 20]. A significant improvement has been made possible by integrating locomotion and manipulation capabilities on a single platform. Mobile bases are highly effective for mobility on their own, while robotic arms are not mobile. A combined system that integrates hexapod mobility with a robotic arm enables end-to-end autonomy. This allows for the sequential execution of multiple capabilities, such as traversing rough terrain and target detection. This is crucial for real-world applications such as planetary exploration or disaster relief scenarios [6, 15, 11]. The role of artificial intelligence is significant in achieving such autonomy, particularly through real-time decision-making. Advanced computer vision models such as YOLOv8, with their lightweight architecture and fast inference speed, enable real-

time detection and tracking. This allows real-world robots to perceive their environment through such intelligent capabilities, thereby closing the gap between sensing and acting. However, most robots are still in siloed environments, where they are either highly effective at mobility, such as legged walkers, or highly effective at manipulation, such as robotic arms. However, intelligent perception capabilities are still lacking. The high cost, computational requirements, and lack of robustness are significant barriers to robots operating effectively.



Fig. 1: Hexapod Structure and Robotic Arm

In this paper, a cost-effective, AI-based, multi-terrain hexapod robot with a 4-DOF robotic arm and a YOLOv8-based vision system is proposed for autonomous movement and object handling. The key features of this proposed hex-pod robot is tripod-walking for terrain movement, uses an NRF24L01 for wireless communication, and performs edge-based object detection using cost-effective hardware like a Raspberry Pi. The proposed hexapod robot's contributions are a cost-effective hardware architecture with a cost of < 20,000, optimized hexapod walking for stability, and AI-based object detection with 85% mAP in real time using edge-based hardware like a Raspberry Pi. The proposed hexapod robot's practical significance lies in the development of cost-effective robots for various applications in developing nations.

## II. LITERATURE REVIEW

Research on hexapod robots has progressed from mechanical design and basic gait generation toward terrain-adaptive locomotion, autonomous perception, and integrated mobile manipulation. Early studies established that six-legged platforms offer higher static stability than wheeled or tracked robots on uneven ground because at least three legs can remain in contact with the terrain during walking. More recent work has focused on adaptive gaits, vision-based navigation, lightweight object detection, and the integration of locomotion with robotic arm manipulation.

### *A. Hexapod locomotion and gait control*

Tedeschi and Carbone [2] reviewed design issues for hexapod walking robots, including leg configuration, actuation, kinematics, gait planning, and control architectures. Their work highlighted the trade-off between mechanical simplicity, payload capacity, stability, and terrain adaptability, providing a foundation for low-cost hexapod designs. Yang [3] proposed fault-tolerant gait planning for a hexapod robot walking over rough terrain. The study showed that redistributing support legs and modifying foot trajectories can maintain stability when one or more legs lose contact or fail, which is important for disaster-response and outdoor field robots. Currie, Beckerleg, and Collins [4] examined the software evolution of a hexapod walking gait and demonstrated that modular gait software can support stable walking while enabling iterative improvements to motion parameters. Their approach is relevant to embedded implementations in which gait routines must be updated without redesigning the complete hardware system. Bai et al. [9] developed a central-pattern-generator-based control method for a curved-leg hexapod robot. Using modified Hopf oscillators, the System generated tripod, quadruped, and wave gaits and achieved smooth transitions between them; both simulations and physical experiments confirmed stable, continuous gait switching. This supports the use of the tripod gait in the proposed robot and suggests that future versions could add gait transitions for more difficult terrain. Zheng, Xu, Tian, and Ding [10] analysed a radially symmetrical hexapod robot with leg–arm integration. They showed that one or more legs can function as manipulators, enabling the robot to grasp objects while maintaining a stable supporting gait. Their work is closely related to the proposed System, although the present design uses a separately mounted 4-DOF arm rather than converting walking legs into manipulators.

Tedeschi and Carbone [2] reviewed design issues for hexapod walking robots, including leg configuration, actuation, kinematics, gait planning, and control architectures. Their work highlighted the trade-off between mechanical simplicity, payload capacity, stability, and terrain adaptability, providing a foundation for low-cost hexapod designs. Yang [3] proposed fault-tolerant gait planning for a hexapod robot walking over rough terrain. The study showed that redistributing support legs and modifying foot trajectories can maintain stability when one or more legs lose contact or fail, which is important for disaster-response and outdoor field robots. Currie, Beckerleg, and Collins [4] examined the software evolution of a hexapod walking gait and demonstrated that modular gait software can support stable walking while enabling iterative improvements to motion parameters. Their approach is relevant to embedded implementations in which gait routines must be updated without redesigning the complete hardware system. Bai et al. [9] developed a central-pattern-generator-based control method for a curved-leg hexapod robot. Using modified Hopf oscillators, the System generated tripod, quadruped, and wave gaits and achieved smooth transitions between them; both simulations and physical experiments confirmed stable, continuous gait switching. This supports the use of the tripod gait in the proposed robot and suggests that future versions could add gait transitions for more difficult terrain. Zheng, Xu, Tian, and Ding [10] analysed a radially symmetrical hexapod robot with leg–arm integration. They showed that one or more legs can function as manipulators, enabling the robot to grasp objects while maintaining a stable supporting gait. Their work is closely related to the proposed System, although the present design uses a separately mounted 4-DOF arm rather than converting walking legs into manipulators.

### *B. Vision, obstacle avoidance, and AI perception*

Billah [5] developed a walking hexapod robot for disaster recovery, focusing on terrain negotiation and navigation algorithms. The work demonstrated the suitability of hexapod platforms for search-and-rescue scenarios where wheeled vehicles cannot operate reliably. Hu et al. [19] proposed a self-supervised, biologically inspired obstacle-avoidance

method for a hexapod walking robot. Their controller combined an insect-inspired LGMD visual network, a CPG locomotion controller, and an LSTM-based decision layer; this System was validated in both simulation and real cluttered environments. Thus, an alternative to YOLO-based detection was presented for cases where computational resources are highly constrained. Bagheri et al. [20] implemented resource-efficient bio-inspired visual processing on the HECTOR hexapod robot. Their System-on-Chip module extracted nearness information from visual motion and enabled navigation to target positions without collisions, while consuming far fewer resources than CPU- or GPU-based implementations. The proposed YOLOv8 system differs in that it provides semantic object labels and bounding boxes, which are more useful for target-specific pick-and-place operations.

Zhao, Chai, Gao, and Qi [21] proposed an obstacle-avoidance and motion-planning scheme for the Octopus-III hexapod robot. Their work showed that combining perception with planned footholds and body motion improves navigation in obstacle-rich environments. The proposed robot uses a simpler rule-based avoidance strategy, making it suitable for a low-cost embedded platform. Recent vision-guided hexapod research has also used lightweight YOLO models [23,24] for autonomous object localization. One System combined a USB camera, ultrasonic sensors, ESP32 servo control, Raspberry Pi-based YOLO inference, and a wave gait to locate and approach targets indoors; it achieved high detection precision and repeatable navigation in test trials. This is directly comparable to the proposed robot's Camera–YOLOv8–decision-making pipeline.

*C. YOLOv8 and robotic manipulation*

YOLOv8 has become attractive for robotic applications because it supports real-time inference and can be deployed in lightweight versions on edge devices. In an augmented-reality-assisted robotic arm system, a YOLOv8n model detected objects in real time and provided target coordinates for autonomous arm navigation and grasping. This validates the use of YOLOv8 for the proposed robot's detect-and-manipulate workflow. [23] An EPSO-based robotic-arm framework, EPSO-TPPP-YOLOv8, used YOLOv8 for object detection in fruit-picking tasks. The authors adopted the nano version for hardware deployment and reported near-perfect detection accuracy in unobstructed conditions, although inference speed on the embedded platform was limited. This highlights the importance of balancing model accuracy, frame rate, and edge computing capacity—exactly the trade-off that the proposed Raspberry Pi-based design addresses.

A lightweight YOLOv8 variant, YOLOv8-MCDE, was proposed [23] for small-instrument detection by inspection robots. By using a MobileNetV3 backbone, cross-scale feature fusion, deformable large-kernel attention, and EIOU loss, the model improved precision while reducing computational cost and model size. Such optimization techniques could improve the proposed System's detection performance on low-power hardware. Guo [24] reviewed lightweight YOLO algorithms for vision-based robotic arms and categorized improvements into backbone lightweighting, structural optimization, and model compression. The review emphasized that detection accuracy must be coordinated with grasp execution and real-time constraints. The proposed System currently uses the standard YOLOv8; future work could adopt these compression methods to enable faster onboard inference.

The proposed work addresses this gap by integrating a six-legged tripod-gait hexapod, a 4-DOF robotic arm, YOLOv8-based object detection, Raspberry Pi edge inference, Arduino Mega low-level servo control, and NRF24L01 wireless communication in a single low-cost platform. Unlike prior systems that focus separately on locomotion or manipulation, the proposed robot supports rough-terrain movement, obstacle avoidance, target detection, and pick-and-place operations within a single embedded architecture.

It presents a novel approach to planning the foot trajectory of a legged robot during walking. The main objective of the study is to improve the stability and adaptability of legged robots when traversing complex terrain, including slopes, uneven surfaces, and obstacles. In real-world environments, robots often encounter unpredictable ground conditions, necessitating intelligent trajectory-planning methods that enable them to adjust their movement safely and efficiently. To address this problem, the authors designed a specialized foot-sensing structure integrated into the robot's foot. The sensing mechanism uses a small push-button-type sensor embedded in the foot structure. When the robot's foot touches the ground, the button is pressed, and the contact signal is transmitted to the robot's controller. This provides accurate information about the exact moment when the foot lands on the ground. By detecting this landing event, the robot can adjust its motion in real time and maintain better balance during locomotion. Based on landing data obtained from the sensor, the authors proposed a trajectory-generation technique known as the three-element trajectory-planning method. This method divides the walking motion into phases and plans the robot's foot motion using three key elements: Start point of the support phase, End point of the support phase, and Joint angle change during the swing phase. The support phase trajectory is planned in Cartesian space using smooth curves. The swing phase is planned in joint space using angle functions. This combination makes movement more stable and adaptive. They tested the method on their own hexapod robot using a triangle gait. The results show that the robot can walk stably even in complex environments. By sensing when the foot touches the ground and carefully planning its motion, the robot can walk more safely and adapt to uneven terrain.

It introduces a new mathematical method, Directed Angular Restitution (DAR), to improve inverse kinematics for hexapod robots. Inverse kinematics means calculating joint angles needed to move the leg to a specific position. Traditional methods use trigonometric functions like sine and cosine. However, these methods sometimes cause errors such as sign confusion, limited angular range, and instability. The DAR method solves these problems by separating angle calculation from direction calculation. It uses special elimination and selection vectors to calculate angles between vectors in 3D space. It also introduces "spin functions" to clearly determine the rotation direction (clockwise or counterclockwise). This makes motion smoother and avoids sudden jumps in angle values. The method was tested on a hexapod model and demonstrated higher accuracy and stability than traditional inverse kinematics. DAR improves inverse kinematics by making angle calculations more stable, accurate, and smooth for omnidirectional movement.

It describes how to design and build a hexapod robot using an Arduino Mega 2560. The robot has 18 servo motors, with three motors in each leg. Because the Arduino does not have

enough output pins to control all motors directly, the authors used two PCA9685 servo driver boards connected via I2C. The robot body is made of 5-mm-thick acrylic. Each leg has three degrees of freedom, allowing it to move up and down and forward and backward. The robot uses the tripod gait method for walking. In this method, three legs move together while the other three support the body. The authors tested the robot's movement. The robot moved forward at about 6 cm per second. It rotated 90 degrees at around 18 degrees per second. It could also rotate 180 degrees at a similar speed. This research shows that a hexapod robot can be built at low

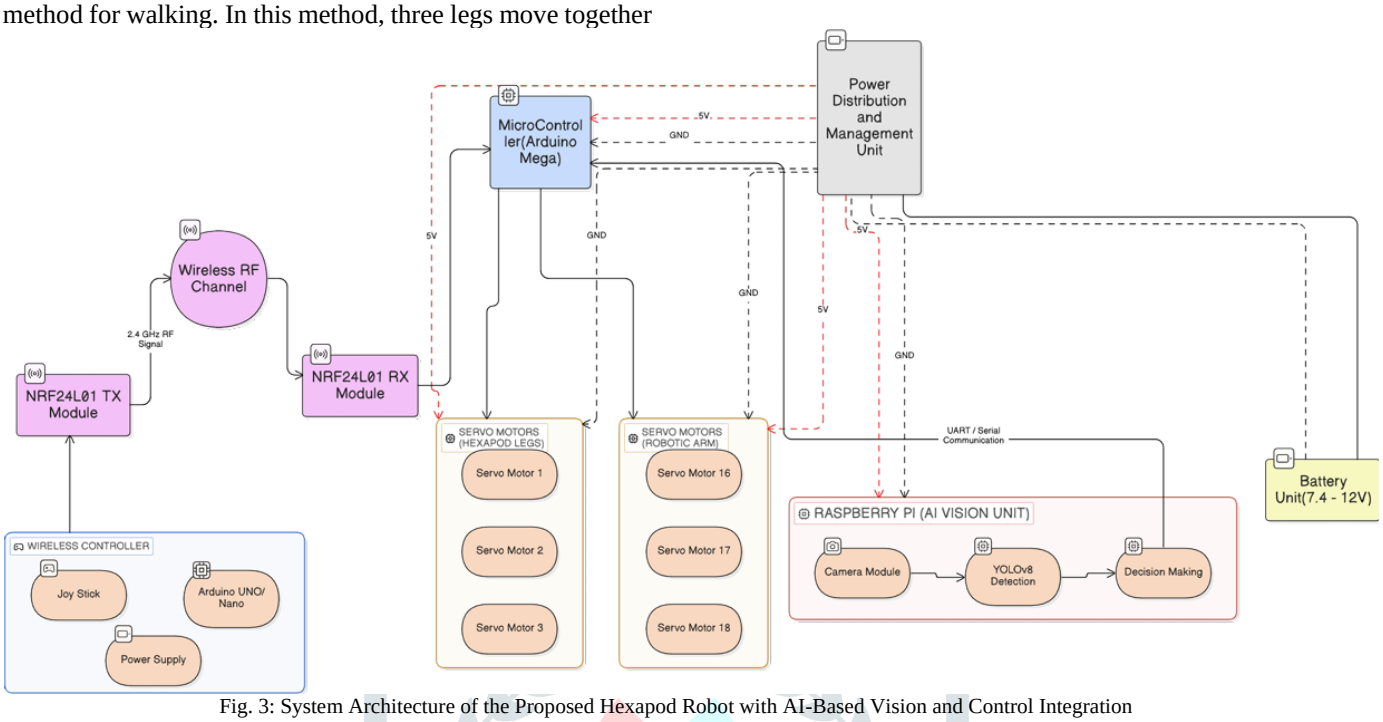


Fig. 3: System Architecture of the Proposed Hexapod Robot with AI-Based Vision and Control Integration

cost using Arduino and simple components. It demonstrates that proper programming and motor control can produce stable, controlled walking.

## III. METHODOLOGY

The proposed System follows a perception-decision-action process to achieve autonomous navigation and object manipulation. This methodology includes three main parts: system architecture, hardware design, and software design.

*System Architecture:* The complete architectural design of the suggested System is framed around the perception-decision-action chain concept, where AI, embedded control, and actuation are incorporated as one entity, as illustrated in Fig. 3. The System is divided into four Phases:

*Vision Module:* This module consists of a camera connected to a Raspberry Pi, with YOLOv8 used as the deep learning model for object detection. The processed images will enable detection of obstacles and targeted objects. Depending on the results, the decision-making process will produce high-level commands such as move, avoid obstacles, or handle specific objects.[1]

*Control Unit:* This module consists of an Arduino Mega + NRF24L01 with PA+LNA. The control board is based on the Arduino Mega board. Low-level tasks, such as handling movement, have been carried out with the Arduino Mega. The wireless connection is implemented using an NRF24L01 module, thereby enabling remote manual control via a joystick. The UART interface sends high-level commands from the decision-making algorithm to the Arduino board.

*Actuation System*: This System consists of a hexapod robot and a multi-DOF robotic arm. For actuators, the System uses a hexapod robot for locomotion and a multi-DOF robotic arm for object manipulation. The hexapod robot moves around in tripod gait mode, while the robotic arm enables pick-and-place operations.

*Servo System:* This module consists of movement and arm action. Multiple servos have been utilized to control the leg joints and the robotic arm. This means controlling servos using PWM signals through the Arduino Mega controller.

*Hardware Design*: The hardware architecture for the development of this project is created to make sure that the locomotion remains steady and the objects can be manipulated reliably.

There are six legged on the platform, with each leg having three DOF: coxa, femur, and tibia joints as shown in the Fig. 4. Each of the joints is actuated using high torque 18 MG996r metal gear servo motors for Robotic Arm and 7 servo motor for robotic arm which include 3 MG90s metal gear servo motor and rest 4 MG996r metal gear servo motors, which enables the precision of control and coordination during the movement of both Hexpod Robot and Robotic Arm with is



Fig. 4. 3-DOF Hexapod Leg Design showing Coxa, Femur, and Tibia Joints

on top of the Hexapod Robot. Tripod gait algorithms ensure that three legs touch the ground at all times, and 3-leg walking contributes to greater stability on uneven surfaces [10, 17].

A four-degree-of-freedom robotic arm is mounted on top of the hexapod structure. This robotic arm is used to manipulate various objects by picking them up and placing them back, as shown in Fig. 5, using the servo motors as the drive elements. The structures of both the platforms and the robotic arm itself are created using 3D printing. Parts of the robot are constructed using 3D printing with lightweight materials such

as PLA+. With this method, fast prototyping is possible, with lower manufacturing costs.

and easier design iterations. The printed parts are optimized to achieve an appropriate strength-to-weight ratio. There is a need to design these parts with a high strength-to-weight ratio [7, 16]. A comprehensive summary of the 3D-printed components, including fabrication time and 3D part details, is provided in Table 1

Table 1: SUMMARY OF 3D PRINTED COMPONENTS OF THE HEXAPOD ROBOT SYSTEM

| Component | Quantity | Total Time | Function |
|---|---|---|---|
| Hexapod Lower Base | 1 | 15 hrs | Central structural frame supporting all modules |
| Leg A | 6 parts | 16 hrs | Main leg structure for movement and balance |
| Leg B | 6 parts | ∼20 hrs | Joint connection enabling bending and stability |
| Leg C | 6 parts | ∼27 hrs | Lower leg support improving walking stability |
| Shafts | 20 parts | 3 hrs | Enables smooth rotation and joint motion |
| Remote Controller | 1 | 8h 55m | Protects and stabilizes electronics |
| Upper Body | 1 | 8h 30m | Top frame for structural support |
| Arm (A+B) | 2 parts | 13h 30m | Main robotic arm links for lifting and movement |
| Arm Base + Disc | 2 parts | 6h 47m | Provides rotational base for arm |
| Arm Base Body | 1 | 3h 21m | Structural support for arm mounting |
| Gripper | 1 | 3h 33m | End-effector for grasping objects |
| Base Shaft | 1 | 1h 30m | Central support for mechanical stability |

A dual 20A buck converter system is used to ensure stable power distribution for high-current servos and control units. Power management is done through the use of a central power distribution system, whereby the power source from batteries with voltages ranging from 7.4-12V is controlled by buck converters, which ensure a constant level of power supply to the control electronics and the servos. In general, the hardware design has considered mechanical stability, energy conservation, and flexibility for easy integration into control and vision AI systems.

Software Design: The software architecture of the proposed System has been designed to enable the whole System to control these three activities simultaneously: Locomotion, Communication, and the AI Module. The System follows a modular approach, in which individual functionalities have been incorporated separately. The locomotion pattern of the hexapod robot is based on the tripod gait algorithm, in which only three legs are grounded at a time while the other three move forward or backward. This gait algorithm provides static stability during walking over multi-terrain. The gait cycle is executed by defining the trajectory of joint angles, which are converted to PWM values [4, 5, 12].
Communication between the user interface and the robot occurs via wireless data transmission using an NRF24L01 with a PA+LNA, powered by a power adaptor. The NRF24L01 is a radio-frequency communication device operating in the 2.4 GHz frequency band. Data from the joystick, via the Arduino Mega microcontroller, is then transmitted to the robot as control messages. The receiver end, which involves an Arduino Mega, will then decode the signals into relevant motion commands [8]. This AI module is based on the YOLOv8 implementation, which is used for object detection in a deep learning framework running on a Raspberry Pi in real time. The visual composition, or any frames obtained from the camera module, is analyzed to determine whether it is an obstacle or a target. This data is used to generate control signals that provide directions such as moving forward, avoiding obstacles, or picking up an object with the gripper. These high-level commands are communicated to the Arduino Mega via UART/serial communication [2, 3]. The steps shown in Fig. 6 are explained below.

1. Image Acquisition: The camera module is connected to the Raspberry Pi, which captures real-time frames of the surroundings
2. Object Detection: The captured images are processed using a YOLOv8 model deployed on the Raspberry Pi to detect obstacles.
3. Object Classification: The detected objects are divided into two categories: either obstacles or a target object
4. Decision Making: This rule-based decision algorithm controls instructions such as movement, object avoidance, and object pickup and placement.
5. Command Transmission (UART): These commands are sent from the Raspberry Pi to the Arduino Mega board via serial (UART) communication.
6. Wireless Control Integration (NRF24L01): The control commands are sent via the NRF24L01 with PA+LNA radio transceivers, and the Arduino Mega receives the signals to drive the servo motors to move forward, backward, left, or right.
7. Control Processing (Arduino Mega): The Arduino Mega serves as the brain of the Hexapod Robot, processing both artificial intelligence instructions and wireless signals and sending the corresponding control signals.
8. Servo Signal Generation: The Arduino Mega generates PWM signals to control all the servo motors for both hexapod and robotic arm movement.
9. Execution: Servo motors control the legs of the hexapod and the robotic arm to perform multiple operations, such as walking, avoiding obstacles, and picking up and placing objects.

The integrated robotic arm can be used to clear lightweight obstacles, grasp and deliver emergency supplies, or collect samples from hazardous zones [13]. • Military Surveillance: Its low-noise operation, stable gait, and ability to maintain balance while moving enable effective surveillance in forested, mountainous, or urban rubble areas. The robot can perform inspection, maintenance, and material handling tasks in factories, warehouses, or assembly lines, especially in constrained or elevated spaces. The design minimizes human exposure to hazards and enables continuous monitoring and data collection in extreme conditions. The robot can navigate uneven agricultural fields, crop rows, or sloped terrain where wheeled robots struggle.

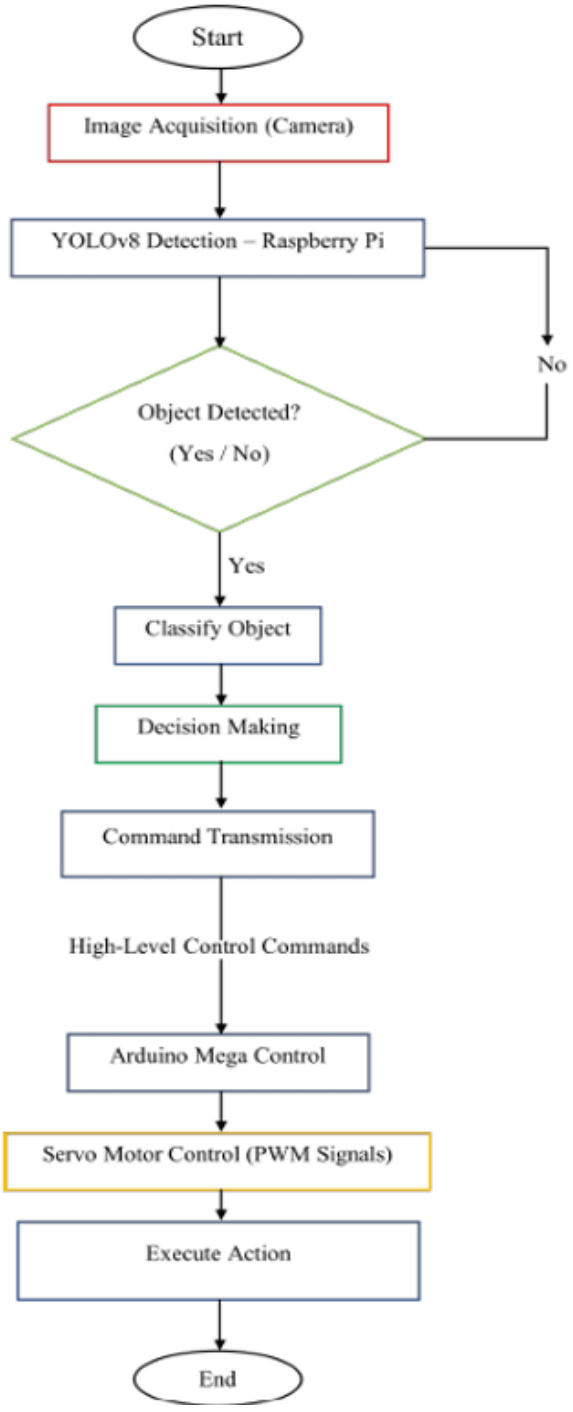


Fig. 6. Methodology Flowchart of the Proposed Hexapod Robot System

## IV. RESULTS

Table 2 demonstrates that this new approach is significantly better than the traditional one in terms of these proposed system benefits. By comparing traditional robots with the new approach, which involves hexapod robots confined to wheel or track movement, the hexapod robot allows for more flexible and stable movement on multiple surfaces; moreover, adding a robotic arm on top of the hexapod robot

TABLE 2: COMPARISON BETWEEN TRADITIONAL ROBOTS AND PROPOSED SYSTEM

| Feature | Traditional Robots | Proposed System |
|---|---|---|
| Mobility | Limited (Wheels/Tracks) | Multi-terrain (Hexa-pod) |
| Manipulation | Not Available | Robotic Arm (4-DOF) |
| AI Vision | Not Integrated | YOLOv8-based De-tection |
| Autonomy | Low / Manual | Semi-Autonomous |
| Communication | Wired / Basic RF | NRF24L01 + UART |
| Control System | Single-layer | Distributed (Rasp-berry Pi + Arduino) |
| Terrain Adapt-ability | Low | High |
| Cost | High | Low (< ₹ 20,000) |

gives the System the ability to manipulate objects, which is not possible with traditional designs. The application of an AI-based vision algorithm (YOLOv8) improves the accuracy of pick-and-place operations and supports semi-autonomous operation. Also, the wireless communication (NRF24L01) with PA + LNA between the robot and the remote control, and the distributed control (Raspberry Pi and Arduino Mega), make the robot more flexible and responsive. Despite these improvements, the proposed System remains less expensive than traditional ones.

## V. CONCLUSION AND FUTURE WORK

This paper presents a novel AI-driven hexapod robot with an integrated robotic arm. The System successfully combines locomotion, perception, and manipulation. Results validate its effectiveness for real-world applications. Fabricated primarily through 3D printing, the robot offers a low-cost, lightweight, and modular design that enables rapid prototyping and easy component replacement. The developed platform demonstrates stable hexapod locomotion on uneven terrain, along with reliable manipulation capabilities, making it suitable for challenging environments. Its applications are in critical domains such as search-and-rescue operations, military surveillance, industrial automation, and hazardous-environment exploration. Future work includes SLAM-based navigation, lightweight AI deployment on edge devices, and enhanced robotic arm precision with feedback systems. Vision-based navigation and mapping will be implemented to enable the robot to traverse unknown, unstructured environments autonomously. The robotic arm will be upgraded with closed-loop control using joint encoders or torque feedback sensors. Lightweight AI models will be deployed for real-time decision-making.

## REFERENCES

[1] Siegwart, Roland, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.

[2] Tedeschi, Franco, and Giuseppe Carbone. "Design issues for hexapod walking robots." *Robotics* 3.2 (2014): 181-206.

[3] Yang, Jung-Min. "Fault-tolerant gait planning for a hexapod robot walking over rough terrain." *Journal of intelligent and robotic systems* 54.4 (2009): 613-627.

[4] Currie, Jonathan, Mark Beckerleg, and John Collins. "Software evolution of a hexapod robot walking gait." *International journal of intelligent systems technologies and applications* 8.1-4 (2010): 382-394.

[5] Ahmed, Mohiuddin, et al. *Walking hexapod robot in disaster recovery: developing algorithm for terrain negotiation and navigation*. INTECH Open Access Publisher, 2010.

[6] Arrigoni, Stefano, Marco Zangrandi, Giovanni Bianchi, and Francesco Braghin. "Control of a hexapod robot considering terrain interaction." *Robotics* 13, no. 10 (2024): 142.

[7] Tedeschi, Franco, and Giuseppe Carbone. "Design issues for hexapod walking robots." *Robotics* 3, no. 2 (2014): 181-206.

.

[8] Arrigoni, Stefano, Marco Zangrandi, Giovanni Bianchi, and Francesco Braghin. "Control of a hexapod robot considering terrain interaction." *Robotics* 13, no. 10 (2024): 142.

[9] Bai, Long, Hao Hu, Xiaohong Chen, Yuanxi Sun, Chaoyang Ma, and Yuanhong Zhong. "Cpg-based gait generation of the curved-leg hexapod robot with smooth gait transition." *Sensors* 19, no. 17 (2019): 3705.

[10] Zheng, Yi, Kun Xu, Yaobin Tian, and Xilun Ding. "Different manipulation mode analysis of a radial symmetrical hexapod robot with leg—arm integration." *Frontiers of mechanical engineering* 17, no. 1 (2022): 8.

[11] Ma, Jiawang, Guanlin Qiu, Weichen Guo, Peitong Li, and Gan Ma. "Design, analysis and experiments of hexapod robot with six-link legs for high dynamic locomotion." *Micromachines* 13, no. 9 (2022): 1404.

[12] Currie, Jonathan, Mark Beckerleg, and John Collins. "Software evolution of a hexapod robot walking gait." *International journal of intelligent systems technologies and applications* 8, no. 1-4 (2010): 382-394.

[13] Ahmed, Mohiuddin, Md Raisuddin Khan, Masum Billah, and Soheli Farhana. *Walking hexapod robot in disaster recovery: developing algorithm for terrain negotiation and navigation*. INTECH Open Access Publisher, 2010.

[14] Rampersad, Akhil, and Bashan Naidoo. "Hexapod Locomotion Across Structured and Unstructured Terrains: A Data-Driven Review of Modeling, Control, and Validation." *Journal of Field Robotics* (2026).

[15] Yang, Jung-Min. "Fault-tolerant gait planning for a hexapod robot walking over rough terrain." *Journal of intelligent and robotic systems* 54, no. 4 (2009): 613-627.

[16] Chen, Chen, Junbo Lin, Bo You, Jiayu Li, and Biao Gao. "Hexapod robot motion planning investigation under the influence of multi-dimensional terrain features." *Frontiers in Neurorobotics* 19 (2025): 1605938.

[17] Bjelonic, Marko, Navinda Kottege, Timon Homberger, Paulo Borges, Philipp Beckerle, and Margarita Chli. "Weaver: Hexapod robot for autonomous navigation on unstructured terrain." *Journal of Field Robotics* 35, no. 7 (2018): 1063-1079.

[18] Zhao, Yue, Xun Chai, Feng Gao, and Chenkun Qi. "Obstacle avoidance and motion planning scheme for a hexapod robot Octopus-III." *Robotics and Autonomous Systems* 103 (2018): 199-212.

[19] Čížek, Petr, and Jan Faigl. "Self-supervised learning of the biologically-inspired obstacle avoidance of hexapod walking robot." *Bioinspiration & biomimetics* 14, no. 4 (2019): 046002.

[20] Meyer, Hanno Gerd, Daniel Klimeck, Jan Paskarbeit, Ulrich Rückert, Martin Egelhaaf, Mario Porrmann, and Axel Schneider. "Resource-efficient bio-inspired visual processing on the hexapod walking robot HECTOR." *Plos one* 15, no. 4 (2020): e0230620.

[21] Xu, Tong, Xiaomin Han, Honghui Liu, and Yuhong Li. "EPSO-based rigid robotic arm for obstacle avoidance object grasping." *Scientific Reports* 15, no. 1 (2025): 39462.

[22] Hyung, Joung-Woo, Woochan Na, Kanghee Won, and Dong-Joo Kim. "Robotic arm control by augmented reality-assisted object detection: J.-W. Hyung et al." *Scientific reports* 15, no. 1 (2025): 35678.

[23] Yang, Tongxin, Ding Ling, Qingwu Shi, and Tianyue Jiang. "YOLOv8-MCDE for lightweight detection of small instruments in complex backgrounds from inspection robots' perspective." *Scientific Reports* 15, no. 1 (2025): 32060.

[24] Mittal, Payal. "A comprehensive survey of deep learning-based lightweight object detection models for edge devices: P. Mittal." *Artificial Intelligence Review* 57, no. 9 (2024): 242.