Robotica (2025), **43**, pp. 4366–4383
doi:10.1017/S0263574725102907

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Design and implementation of an open-source, low-cost remote-controlled tiny humanoid robot for accessible robotics applications

Marcos Eduardo Pivaro Monteiro[1,4], Eduardo Nunes dos Santos[1], Daniel Fernando Pigatto[1], Joilson Alves Junior[1], Leonardo Lira Ramalho[2,4] and Diego Issicaba[3,4]

[1]Federal University of Technology - Paraná, Curitiba, Paraná, Brazil
[2]Federal University of Pará, Belém, Pará, Brazil
[3]Federal University of Santa Catarina, Florianópolis, Santa Catarina, Brazil
[4]INESC P&D Brazil, Santos, São Paulo, Brazil
Corresponding author: Marcos Eduardo Pivaro Monteiro; Email: marcose@utfpr.edu.br

Received: 31 May 2025; **Revised:** 29 August 2025; **Accepted:** 5 November 2025; **First published online:** 1 December 2025

Keywords: humanoid robot; teleoperation; embedded applications; virtual reality; low-cost robotics

Abstract

This paper presents the design and implementation of Jaeger UTFPR, an open-source, low-cost, remote-controlled tiny humanoid robot measuring just 12 cm in height. Developed with a focus on accessibility and affordability, the robot integrates 3D-printed components, cost-effective electronics, embedded systems, and wireless communication to provide real-time audio and video feedback through a virtual reality (VR) interface. Operators control Jaeger UTFPR using a VR headset and motion controllers, enabling immersive telepresence and direct manipulation of the robot's movements. With a total cost of just a few tens of dollars, this innovative solution offers broad applications in education, entertainment, research, and remote inspection, serving as an accessible platform for robotics enthusiasts and developers. Experimental evaluations demonstrate the system's effectiveness in balancing performance and cost, validating its potential as a tool for immersive robotics experiences.

1. Introduction

Robots possess immense potential across various domains, offering transformative capabilities in multiple aspects of human life. In living services, they adopt a friendly role and efficiently perform a wide range of tasks [1, 2]. In the entertainment industry, robots enhance visitor engagement in theme parks, museums, and exhibitions by acting as entertainers, presenters, or guides [3]. They are also transforming education by serving as interactive teaching assistants, delivering personalized learning experiences, engaging students in dynamic activities, and providing tutoring support [4]. Beyond these areas, robots are increasingly deployed in military, industrial, healthcare, and agricultural applications, where they contribute to safety, efficiency, and productivity [1, 5].

Robots stand at the forefront of technological innovation, serving as a driving force behind progress and transformation across various industries. Their continuous development is widely recognized as a key indicator of a nation's technological prowess and commitment to innovation [6, 7]. However, unlocking their full potential requires significant breakthroughs in both software and hardware domains, including enhancements in artificial intelligence, sensor technology, energy efficiency, and mechanical design [8, 9]. Recent efforts in the field of robotics have focused on specialized applications, such as improving robot task scheduling in cluttered environments [10], enhancing dynamic locomotion [11], and developing systems with intelligent behaviors [12].

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

Designing human–machine interfaces for mobile robots is a multidisciplinary challenge that requires considering the limitations of the entire control chain. On the robot's side, constraints include mechatronic limitations and processing power, which can restrict onboard sensory data processing. The communication link between the robot and the operator must account for network bandwidth, latency, and adaptability to dynamic conditions, factors that are particularly critical for real-time applications [13]. On the operator's side, human factors such as cognitive limits, psychology, and ergonomics are crucial alongside hardware and software design requirements [14].

In the literature, the term *teleoperation* commonly refers to the scenario in which a human operator remotely controls a robot to perform tasks, often requiring precision and dexterity in the remote environment [15]. Recent works, such as that by ref. [16], have explored advanced virtual reality (VR)-based teleoperation architectures, while ref. [17] has investigated the use of telepresence robots in social settings, such as elder care. While various interpretations exist, in this work, we focus on teleoperation involving a real environment and a humanoid robot. The teleoperation setup can be understood as an interactive system in which the robot replicates the human's actions to achieve a shared objective [15, 18]. Teleoperating humanoid robots present considerable challenges, particularly due to their mechanical complexity and the dynamic, unstructured nature of remote environments [19]. Nevertheless, the ability to integrate human cognition and robotic physical capabilities makes teleoperation a compelling solution for tasks where full autonomy is not yet viable.

Most robotic development focuses on child-sized to adult-sized robots [9]. In ref. [20], the authors introduce SDR-4X, a small humanoid entertainment robot with enhanced human interaction and home adaptability. Nevertheless, its size exceeds half a meter, highlighting the relatively low attention given to very small robots, compared to larger ones. Cost is another primary challenge in this field. In refs. [21] and [22], the low-cost humanoid robot POPPY has been developed using 3D printing technology. However, despite its affordability compared to traditional humanoid robots, its cost still exceeds several thousand dollars.

To overcome these challenges, a number of works have sought to push the limits of cost-effective humanoid design by using additive manufacturing, modular electronics, and compact mechanical topologies [23, 24]. These efforts, including the development of platforms for dynamic locomotion [25] and those designed for inspection in narrow spaces [26], demonstrate the feasibility of low-cost platforms suitable for research and education. Other designs have focused on specific functionalities like unique locomotion mechanisms [27, 28] or low-cost camera systems for real-time applications [29]. While these works represent significant progress, many still depend on custom components or non-trivial assembly techniques. Mechanical design also plays a critical role in enabling efficient motion and structural simplicity, especially in small-scale robots where space and weight constraints are more pressing [30, 31]. Moreover, high-accuracy kinematic models are essential for precise manipulation, even in serial manipulators [32]. While advanced strategies such as adaptive control for varying terrain [33] and high torque-to-weight hydraulic actuators [34] have been proposed for improving locomotion and joint performance, such techniques remain more applicable to larger or high-end platforms. Nonetheless, these insights inform the broader field and inspire scaled-down adaptations in compact systems.

1.1. Main contributions

In this paper, we present Jaeger UTFPR, a compact, low-cost, headset-controlled humanoid robot that stands just 12 cm tall and can be built for only a few dozen dollars. Its structure is fabricated using 3D-printed parts, which ensures both affordability and ease of maintenance. The robot is designed for versatile use across education, entertainment, and research domains. Its key features are as follows:

- Remote-controlled via a VR headset: Offers an immersive teleoperation experience, allowing intuitive control through head movements;
- 3D-printed structure: All mechanical parts are fabricated using additive manufacturing, promoting rapid prototyping, modularity, and low replacement costs;

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

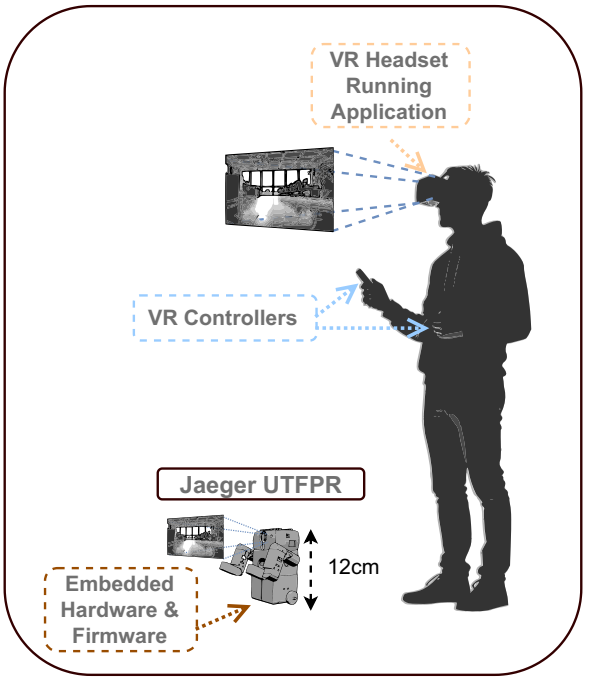


Figure 1. *On the left, Jaeger UTFPR, a small humanoid robot measuring only 12 cm in height. On the right, an operator controlling the robot using a VR headset and its respective controllers.*

- Low cost: The total material and component cost remains within a few dozen dollars, making it accessible to students, educators, and hobbyists. Compared to other platforms such as POPPY [21, 22], it offers a significantly more affordable cost;
- Fully open-source: All hardware schematics, CAD files, and control software are publicly available [35], encouraging replication, customization, and collaborative development by the robotics community.

Thanks to its diminutive scale and immersive control system, Jaeger UTFPR enables users to experience the world from the unique perspective of a 12 cm-tall individual. This feature evokes thematic parallels with the immersive miniaturization explored in the film *Downsizing* [36]. Recent robotics research has explored various aspects of advanced platforms, but none fully integrate the key domains of miniaturization, immersive VR control, and open-source, low-cost design. Some works have focused on miniaturization, developing platforms such as millirobots and bio-inspired systems for narrow spaces [24, 26], as well as insect-sized hexapods [27] and accessible miniaturized multi-robot platforms [37]. A separate body of research has concentrated on VR control, demonstrating immersive teleoperation architectures for remote manipulation [14, 16, 17]. Concurrently, many projects have been dedicated to providing open-source, low-cost solutions, using 3D printing and off-the-shelf components to create affordable platforms for research and education [23–25, 29, 37]. To the best of the authors' knowledge, no single platform combines all three of these domains, making our work a novel solution in the field.

This paper is structured as follows: Section 2 introduces the proposed solution, detailing its hardware, sensors, actuators, and software. Section 3 presents an evaluation of the proposed system, while Section 4 concludes with final remarks.

2. Proposed solution

An overview of the proposed system is presented in Figure 1. On the left, the Jaeger UTFPR robot is shown, measuring only 12 cm in height. It integrates an embedded platform composed of hardware

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

and firmware responsible for capturing data from its sensors and executing actions through its actuators while maintaining communication with a VR headset via Wireless Fidelity (Wi-Fi). On the right, an operator is shown using a VR headset running custom software that displays real-time video and plays audio captured by the embedded system. The operator also holds two hand controllers, which enable control of the robot's actuators, with the controllers' positions continuously tracked by the VR headset. Additionally, the VR headset is responsible for acquiring data from its integrated sensors, including the microphone and controllers, and transmitting this information to the embedded platform. In this way, the operator achieves full control of the miniature robot, experiencing its environment through real-time audio and video feedback, while also being able to manipulate the robot's arms based on the relative positional data of the VR controllers.

The bottom, top, back, front, left, and right views of the robot are presented in Figure 2, highlighting the positions of all actuators and sensors. The actuators include seven stepper motors, which control the movement of the robot's arms and camera; two direct current (DC) motors, responsible for driving the wheels and enabling locomotion; and a speaker for audio output. The sensor array consists of a camera, positioned on the robot's head to capture visual data, and a microphone for acquiring ambient audio. These components are distributed across the robot's structure to ensure efficient movement, environmental perception, and real-time interaction with the operator via the VR system. Additionally, the figure identifies the location of the battery compartment and the power switch.

2.1. Hardware

The hardware consists of a main board and a driver board, as shown in Figure 3. The main board is primarily responsible for all processing operations related to the robot, while the driver board handles the control and state management of all motors. The components were selected to keep costs as low as possible, while also prioritizing parts that are readily available.

2.1.1. Main board

At the core of the main board is the ESP32 module, as shown in Figure 3, positioned on the top layer. This module features an ESP32-D0WDQ6 microcontroller while also integrating a 4 MB SPI flash memory. To support image processing, the main board also includes an APS6404L integrated circuit (IC), which provides 64 Mb of DRAM. The top layer of the board also hosts the speaker and microphone connectors, along with the reset button. On the bottom layer, the camera connector, power input, and board-to-board headers are located. To ensure proper power distribution, both layers include 3.3 V voltage regulators connected via a 0.1 Ω resistor for load balancing. It is worth noting that the ESP32-D0WDQ6 module was chosen for its availability and low cost; however, variations or updated versions of this module can be used with minimal impact on the overall project design.

2.1.2. Driver board

The driver board is responsible for managing and controlling all nine motors in the system. To deliver current to the motors, eight DRV8833 dual H-bridge motor drivers are used – seven dedicated to controlling the stepper motors and one used to control the two DC motors. Each stepper motor requires control of two phases, and since a single DRV8833 can independently control two motor phases or two DC motors, this configuration is both efficient and space-saving.

Given the limited availability of GPIO pins on the microcontroller, three 74HCT259 ICs – 8-bit addressable latches – are employed to manage the motor control signals. Two of these latches are used to drive the motor phases, while the third is dedicated to controlling the enable signals for the drivers. This architecture allows full control of seven stepper motors and two DC motors using only six GPIO pins: three pins (0–7) are used to select which motor configuration to address (in configuration 7, phase

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

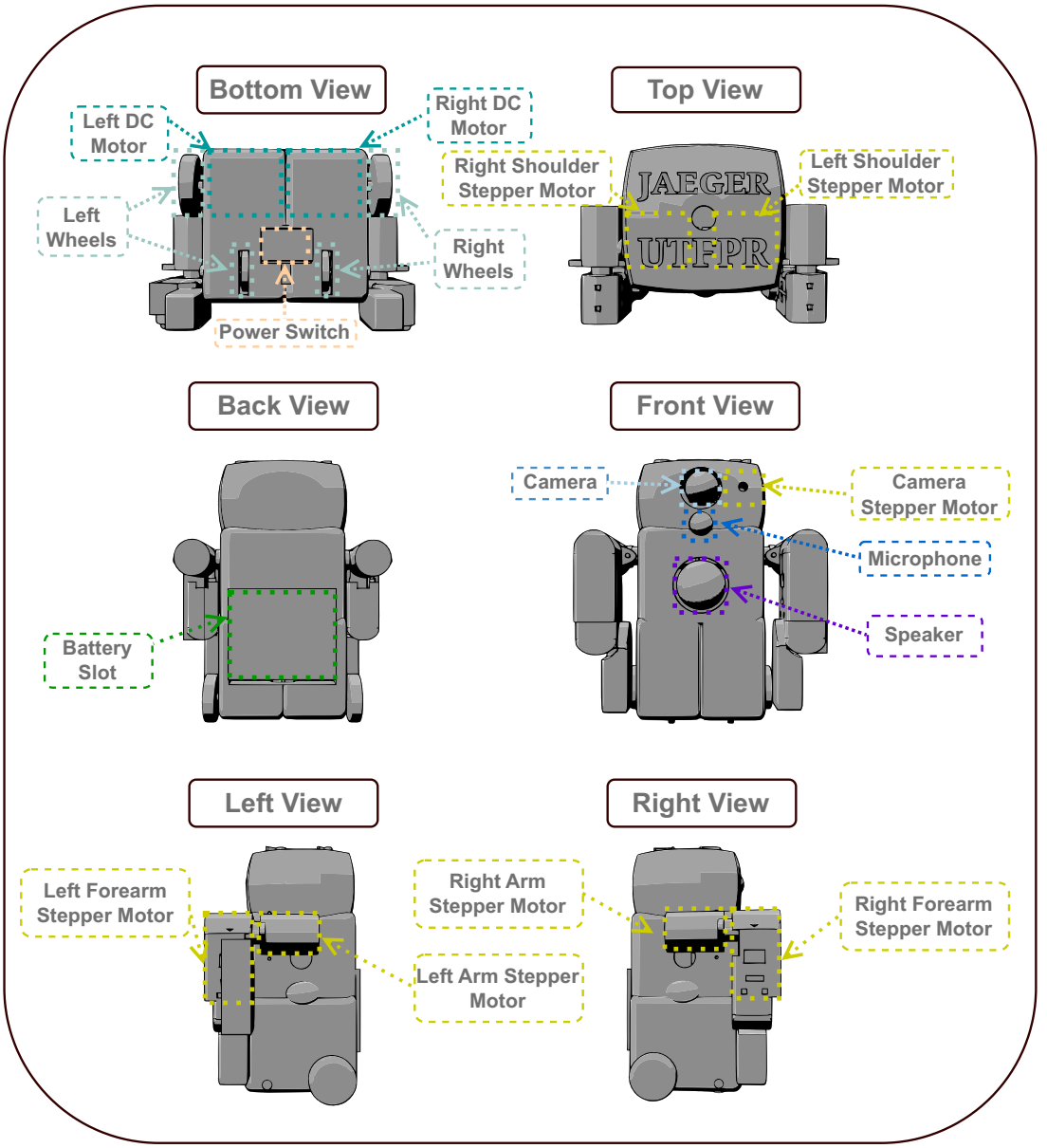


Figure 2. *Bottom, top, back, front, left, and right views of Jaeger UTFPR, indicating the positions of all actuators (seven stepper motors, two DC motors, and one speaker) and sensors (a camera and a microphone).*

1 drives one DC motor and phase 2 the other), and three pins are used to select which of the three latches will be written to (phase 1, phase 2, or enable).

This compact and efficient design enables precise motor control with minimal pin usage. Additionally, motors that are not actively rotating are disabled by the main board to prevent overheating and conserve energy. It is also important to note that each H-bridge on the DRV8833 has two input lines. To enable bidirectional control of each motor phase using only one output from a 74HCT259, a P-Channel Metal-Oxide-Semiconductor Field-Effect Transistor (P-MOSFET) is used to invert the signal from the first

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

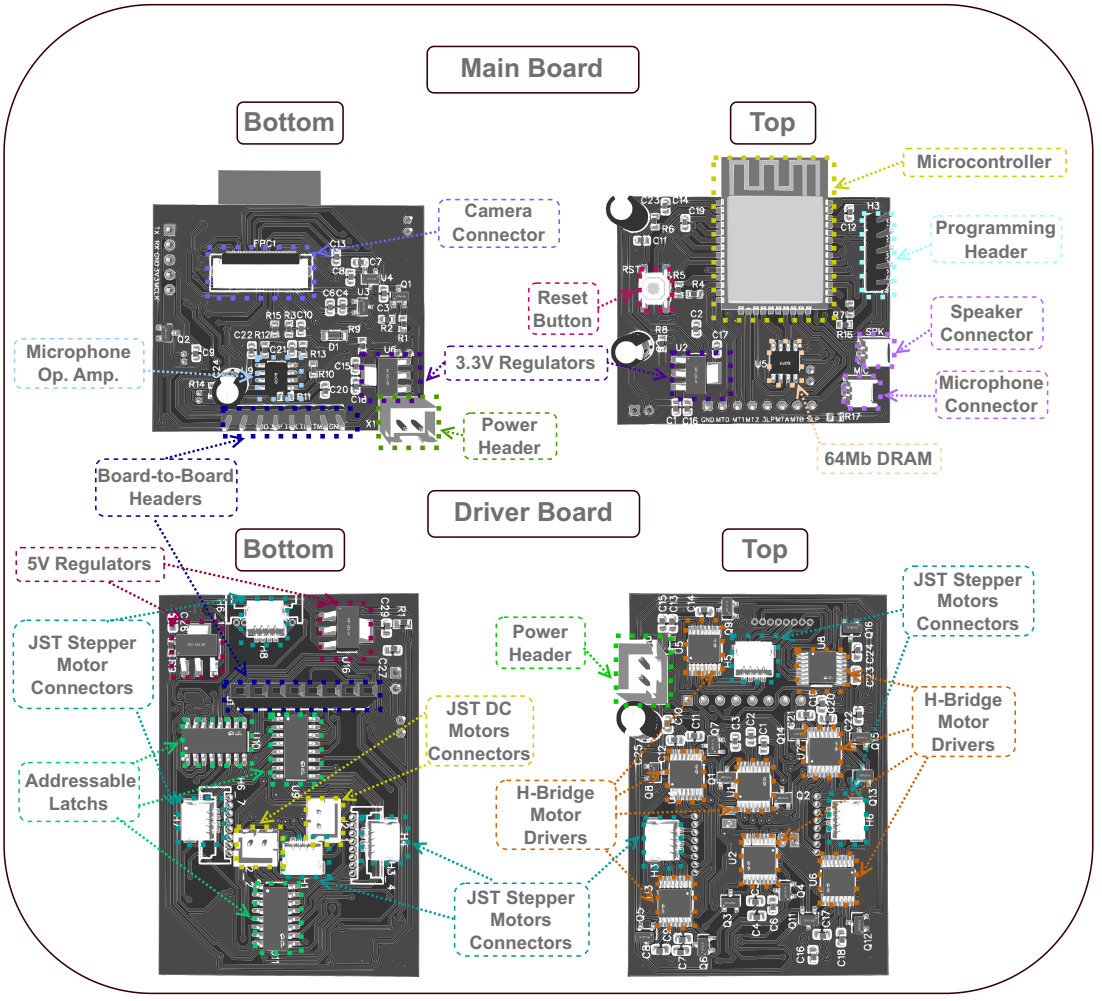


Figure 3. *Top and bottom views of the main and driver PCB boards.*

input and feed it into the second input of the same bridge. This strategy minimizes the number of required control signals while maintaining full bidirectional control over each motor.

To enable wireless operation, the Jaeger UTFPR robot is also equipped with two 16,340 3.7 V battery slots, as shown in the back view of Figure 2. These batteries typically have a capacity of about 1300 mAh each. The batteries are connected in series, supplying power to both the main board and the driver board through their respective power headers shown in Figure 3.

2.2. Sensors

As shown in the front view in Figure 2, the Jaeger UTFPR uses a camera to capture image data from its surroundings. For this purpose, a 24-pin Flexible Printed Circuit (FPC) connector, located on the bottom layer of the main board shown in Figure 3, is used to interface with an OV2640 camera featuring a 160° field of view. This camera is equipped with a sensor array of 1632 columns by 1232 rows, capable of capturing images with resolutions up to 1632 × 1232 pixels. To allow proper positioning within the robot's frame, the camera's FPC cable must be at least 10 cm long. The microphone and speaker are placed as shown in the front view of Figure 2. Directly below the camera, a 9 mm microphone is installed and connected to the Microphone JST connector located on the top layer of the main board, as shown in Figure 3.

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

2.3. Actuators

Below the microphone, a 2 W speaker with a diameter of 2.3 cm is mounted, with its connector also located on the top layer of the main board, as shown in Figure 3. An LM358 operational amplifier, positioned on the bottom layer of the main board, is used to amplify the microphone's signal. The speaker is driven via a digital pin using the Remote Control Transceiver (RMT) peripheral of the ESP32, which emulates Pulse Width Modulation (PWM) behavior. The audio signal is generated at a frequency of 8 kHz with 8-bit resolution per sample. In this setup, each audio sample is mapped to a PWM duty cycle ranging from 0% (0) to 100% (255), where each sample period consists of 255 pulses.

The robot's motor actuators consist of seven stepper motors with 2210 steps per revolution and two N20 DC motors, with their JST connectors located on both the top and bottom layers of the driver board, as shown in Figure 3. Each arm is equipped with three 5 V stepper motors, each having a 15 mm diameter and featuring an embedded 1:110 reduction gear, enabling movement in three degrees of freedom: vertical (up and down), horizontal (forward and backward), and the opening and closing of the forearm. The positioning of each motor is illustrated in the left, right, and top views in Figure 2. Notably, each arm also includes a motor housed within the robot's main body.

The use of low-cost N20 DC motors and small stepper motors was a deliberate choice to provide the robot with a good range of motion without a significant cost increase. The N20 motors provide sufficient torque to move the robot on flat surfaces, while the stepper motors offer precise, repeatable movements for the arms. The trade-off is that these motors have limited speed and torque, which means the robot cannot perform tasks requiring rapid, forceful movements or lift heavy objects. Similarly, the ESP32 microcontroller, while being an excellent low-cost solution for Wi-Fi and processing, is not as powerful as more expensive microcontrollers, which limits the achievable video resolution and frame rate. This highlights the core design philosophy of the proposed platform: balancing performance and cost by making conscious trade-offs in component selection to achieve an accessible, functional system.

2.4. Embedded software

2.4.1. Firmware

The robot's firmware was developed using the Espressif Internet of Things Development Framework, which offers a comprehensive set of tools and libraries to support application development on ESP32-based hardware. The firmware is responsible for real-time tasks such as image acquisition, audio processing, motor control, and wireless communication.

Image data is acquired from the OV2640 camera, with frame rates varying from 15 to 60 fps depending on the selected resolution. Higher resolutions impose a significant processing load on the ESP32 and result in larger image files, which in turn decrease the transmission rate to the VR headset. As shown in Figure 4, multiple ESP32 peripherals are utilized: Inter-Integrated Circuit (I^2C) for camera communication, Universal Asynchronous Receiver/Transmitter for programing and debugging, Analog-to-Digital Converter (ADC) for audio sampling, timers for stepper motor control, RMT (Remote Control) for sound playback, and Wi-Fi for wireless data exchange.

Audio sampling is performed at 24 kHz via the ADC. To reduce high-frequency noise, every three samples are averaged to produce an effective 8 kHz signal. The resulting 8-bit samples are stored in a circular buffer and transmitted in chunks of 4000 samples (equivalent to 500 ms of audio) to the VR headset. Sound playback is handled using the RMT peripheral, which is typically employed for protocol timing but is leveraged here for generating precise PWM signals. Unlike timer-based solutions, RMT offers superior timing accuracy with minimal interference from the main application thread. For each 500 ms block of received audio (4000 samples), the RMT generates a PWM signal composed of 256 high and low duty cycle segments to reconstruct the analog waveform.

Motor control is managed via a queue that stores successive target angles received from the VR headset. The system controls seven stepper motors, each updated using one of two hardware timers. Motors are actuated at speeds of up to 800 steps per second. Given a resolution of 2210 steps per revolution, this equates to a maximum angular velocity of approximately 130°/s per motor. Importantly, each timer can

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

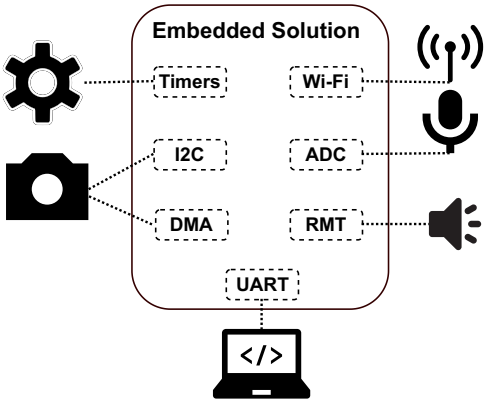


Figure 4. *Peripheral usage.*

operate different motors independently. The firmware also handles concurrent transmission of image data over Wi-Fi and the bi-directional streaming of audio data. While images are transmitted as they become available, audio data is sent in fixed-size chunks of 500 ms.

2.4.2. VR software

To complement the embedded firmware and deliver a complete solution, a custom VR application was developed. This application captures motion data from the VR controllers, computes their relative positions, and calculates the corresponding kinematics, which are then transmitted to the robot in real time. It also captures audio from the VR headset's built-in microphone at an 8 kHz sampling rate, applies 8-bit quantization, and sends the data to the robot for playback. Simultaneously, it receives audio captured by the robot's microphone and plays it back through the VR headset, enabling two-way audio communication.

The application also handles the reception and display of image data transmitted by the robot. This visual feedback is aligned within the user's field of view. To ensure accurate arm movement representation, the VR application includes a calibration feature that allows users to define the relative position of their hands while in a stationary pose. Without this step, reliable inverse kinematics calculations would not be feasible. For communication between the VR headset and the robot's firmware, the solution supports both Transmission Control Protocol (TCP) – a connection-oriented protocol that ensures reliable, ordered data transmission – and User Datagram Protocol (UDP) – a connectionless protocol optimized for low-latency, real-time data exchange.

2.5. Kinematics

To accurately replicate the operator's arm movements, the robot's arm utilizes both forward and inverse kinematics. These techniques enable the computation of joint angles for the three actuated degrees of freedom in the arm, either by exhaustively minimizing positional error across all angle combinations or by directly solving the inverse kinematics equations for a given end-effector target. In this work, both approaches are supported; however, due to their computational demands, all calculations are offloaded to the VR headset rather than performed onboard the robot.

The overall kinematics of the robot's arm are represented by a transformation matrix $\mathbf{K}$, which results from the successive application of rotation and translation matrices. These matrices describe the spatial relationship between the joints and segments of the robot arm in 3D space. Below is a breakdown of each transformation matrix:

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

1. **Rotation Matrix about the x-axis $\mathbf{R}_x$**:

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_1) & -\sin(\theta_1) & 0 \\ 0 & \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{1}$$

   Applies a rotation of angle θ_1 around the x-axis, corresponding to the shoulder pitch movement.

2. **Rotation Matrix about the z-axis $\mathbf{R}_z$**:

$$\mathbf{R}_z = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2}$$

   Represents a rotation by θ_2 around the z-axis, corresponding to shoulder roll.

3. **Translation Matrix along the y-axis $\mathbf{T}_{y-\text{arm}}$**:

$$\mathbf{T}_{y-\text{arm}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\mathcal{D}_{s-e} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

   Translates the coordinate frame by a distance $\mathcal{D}_{s-e}$ (shoulder to elbow) along the negative$y$-axis.

4. **Rotation Matrix for the forearm $\mathbf{R}_{x-\text{forearm}}$**:

$$\mathbf{R}_{x-\text{forearm}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_3) & -\sin(\theta_3) & 0 \\ 0 & \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4}$$

   Rotates the frame by angle θ_3 around the forearm's x-axis, corresponding to elbow pitch flexion.

5. **End-effector Position Vector $\mathbf{p}$**:

$$\mathbf{p} = \begin{pmatrix} 0 \\ -\mathcal{D}_{e-h} \\ 0 \\ 1 \end{pmatrix}. \tag{5}$$

   Defines the hand's position in homogeneous coordinates, where $\mathcal{D}_{e-h}$ is the distance from elbow to hand.

To align with the actual mechanical configuration of the robot, the angles used in the above matrices are adjusted to match the motor rotation directions:

$$\theta_{M1} = -\theta_1, \quad \theta_{M2} = \theta_2, \quad \theta_{M3} = -\theta_3. \tag{6}$$

The final kinematic transformation matrix $\mathbf{K}$ is computed as a product of the individual transformations applied to the end-effector vector $\mathbf{p}$, being given by

$$\mathbf{K} = R_x \cdot R_z \cdot T_{y-\text{arm}} \cdot R_{x-\text{forearm}} \cdot \mathbf{p}. \tag{7}$$

Each matrix multiplication incrementally updates the coordinate frame, accounting for the robot's joint angles and link lengths. This sequential transformation yields the final position of the end-effector

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

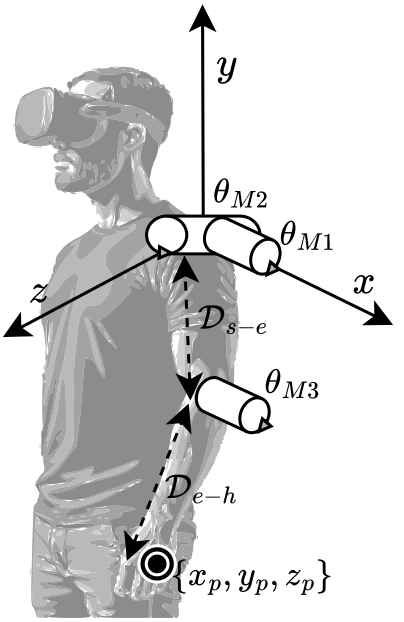


Figure 5. *Perspective view of the humanoid's upper-body kinematics. The diagram shows the global coordinate axes, joint angles (θ_{M1}, θ_{M2}, θ_{M3}), segment lengths ($\mathcal{D}_{s-e}$, $\mathcal{D}_{e-h}$), and hand position components (x_p, y_p, z_p) relative to the VR operator.*

(hand) in the robot's base reference frame. After computing the full transformation chain, the resulting position vector **K** can be analytically expressed as:

$$\mathbf{K} = \begin{bmatrix} \cos(\theta_{M3})\sin(\theta_{M2})\mathcal{D}_{e-h} + \sin(\theta_{M2})\mathcal{D}_{s-e} \\ (\cos(\theta_{M1})\cos(\theta_{M2})\cos(\theta_{M3}) - \sin(\theta_{M1})\sin(\theta_{M3}))\mathcal{D}_{e-h} - \cos(\theta_{M1})\cos(\theta_{M2})\mathcal{D}_{s-e} \\ \cos(\theta_{M2})\sin(\theta_{M1})\mathcal{D}_{s-e} - (-\cos(\theta_{M2})\cos(\theta_{M3})\sin(\theta_{M1}) - \cos(\theta_{M1})\sin(\theta_{M3}))\mathcal{D}_{e-h} \\ 1 \end{bmatrix}. \quad (8)$$

This vector's components correspond to the position of the robot's hand in three-dimensional space: the first element represents the x-axis position (x_p), the second the y-axis position (y_p), and the third the z-axis position (z_p). Therefore, the vector provides a complete mathematical description of the hand's location based on the motor angles and the lengths of the arm segments, effectively modeling the forward kinematics of the robotic arm.

To aid in visualizing the kinematic behavior, Figure 5 shows a perspective view of a person wearing a VR headset. The figure depicts the global coordinate axes (x, y, and z), the hand position components (x_p, y_p, z_p), and the joint rotation angles θ_{M1}, θ_{M2}, and θ_{M3}. These joint angles are represented using cylindrical symbols with arrows indicating the positive direction of rotation. Specifically, θ_{M1} and θ_{M2} are located at the shoulder and correspond to pitch and roll movements, respectively, while θ_{M3} represents the elbow pitch. Additionally, the distances from the shoulder to the elbow ($\mathcal{D}_{s-e}$) and from the elbow to the hand ($\mathcal{D}_{e-h}$) are annotated to provide spatial context for the arm's segment lengths.

As for the inverse kinematics, from Eq. 8, assuming normallized equal proportions between the shoulders and elbow, and elbow and hands, such that $\mathcal{D}_{s-e} = \mathcal{D}_{e-h} = 1$, we have that the estimated motor angles $\theta_{M1}^{\cdot}$, $\theta_{M2}^{\cdot}$, and $\theta_{M3}^{\cdot}$ are given by

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

$$\theta_{M1}^{'} = \frac{180}{\pi}\text{ArcTan}\left[\frac{-32c_5y_p\left(x_p^2+y_p^2+z_p^2\right)+\sqrt{-\left((c_1+c_2-c_4)c_6\right)+1024c_5^2\,y_p^2\left(x_p^2+y_p^2+z_p^2\right)^2}}{c_6},\right.$$
$$-\frac{-c_5\left(c_6-64y_p^2\right)\left(x_p^2+y_p^2\right)+c_5\left(c_3-c_6\right)z_p^2}{2z_pc_6}$$
$$\left.-\frac{y_p\sqrt{-\left((c_1+c_2-c_4)c_6\right)+1024c_5^2y_p^2\left(x_p^2+y_p^2+z_p^2\right)^2}}{z_p(c_3+c_4)}\right], \tag{9}$$

$$\theta_{M2}^{'} = \frac{180}{\pi}\,\text{ArcTan}\left[c_5, \frac{2x_p}{x_p^2+y_p^2+z_p^2}\right], \tag{10}$$

$$\theta_{M3}^{'} = \frac{180}{\pi}\,\text{ArcTan}\left[\frac{1}{2}\left(-2+x_p^2+y_p^2+z_p^2\right),\right.$$
$$\frac{1}{4(c_3+c_4)c_6z_p}\left\{c_4c_6x_p^2y_pc_5+2c_4c_5y_p\left(y_p^2+z_p^2\right)\left(c_6-64\left(x_p^2+y_p^2+z_p^2\right)\right)\right.$$
$$+4c_6\left(y_p^2+z_p^2\right)\sqrt{-\left((c_1+c_2-c_4)c_6\right)+1024c_5^2y_p^2\left(x_p^2+y_p^2+z_p^2\right)^2}$$
$$\left.\left.+c_3\left(c_6x_p^2y_pc_5+2c_5y_p\left(y_p^2+z_p^2\right)\left(c_6-64\left(x_p^2+y_p^2+z_p^2\right)\right)\right)\right\}\right], \tag{11}$$

where ArcTan$[x, y]$ denotes the arctangent function taking into account which quadrant the point (x, y) is in, and the intermediate variables c_1 through c_6 are defined as

$$c_1 = 16\left(x_p^2+y_p^2+z_p^2\right)^2, \tag{12}$$
$$c_2 = -64\,x_p^2, \tag{13}$$
$$c_3 = 64\,y_p^2, \tag{14}$$
$$c_4 = 64\,z_p^2, \tag{15}$$
$$c_5 = \frac{\sqrt{c_2+c_1}}{\sqrt{c_1}}, \tag{16}$$
$$c_6 = c_3+c_4. \tag{17}$$

3. Evaluation

3.1. Kinematics and inverse kinematics evaluation

In order to validate the correctness of the kinematic and inverse kinematic models, we conducted a numerical evaluation using representative joint configurations. In Table I, we begin by applying the actual motor angles (θ_{M1}, θ_{M2}, and θ_{M3}) to Eq. (8) to compute the corresponding positional values x_p, y_p, and z_p. These positional values are then used as input for the inverse kinematics equations Eqs. (9), (10), and (11) to calculate the estimated motor angles ($\theta_{M1}^{'}$, $\theta_{M2}^{'}$, and $\theta_{M3}^{'}$). A comparison of the original and computed angles demonstrates that the inverse kinematics equations accurately reproduce the original joint configurations, validating their correctness.

Note that both forward and inverse kinematics can be used to estimate the correct joint angles. Forward kinematics achieves this by performing an exhaustive search through possible angle combinations and comparing the resulting end-effector position to the desired coordinates (x_p, y_p, z_p). In contrast,

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

Table I. *Inverse kinematics results for $\theta_{M1}^{\cdot}$, $\theta_{M2}^{\cdot}$, and $\theta_{M3}^{\cdot}$.*

Point	$\theta_{M1}[^\circ]$	$\theta_{M2}[^\circ]$	$\theta_{M3}[^\circ]$	x_p	y_p	z_p	$\theta_{M1}^{\cdot}[^\circ]$	$\theta_{M2}^{\cdot}[^\circ]$	$\theta_{M3}^{\cdot}[^\circ]$
1	0	0	0	0	−2	0	0	0	0
2	90	0	0	0	0	2	90	0	0
3	180	0	0	0	2	0	180	0	0
4	90	90	0	2	0	0	90	90	0
5	90	90	90	1	1	0	90	90	90
6	−90	90	90	1	−1	0	−90	90	90
7	90	45	0	1.4142	0	1.4142	90	45	0

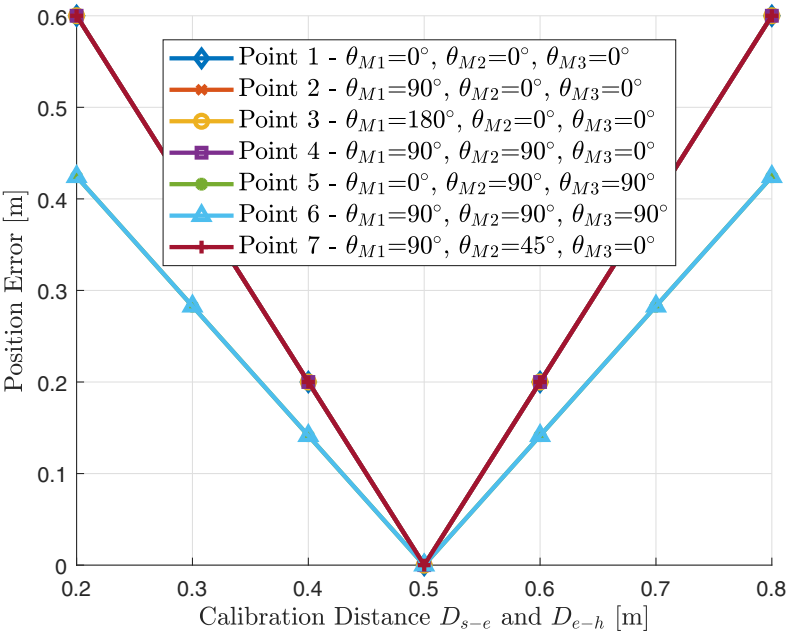


Figure 6. *Position error [m] versus miscalibration of $D_{s\text{-}e}$ and $D_{e\text{-}h}$ [m] (0.5 = true value) for various joint angle combinations.*

inverse kinematics directly computes the angles by substituting the desired position into the inverse kinematic equations. Naturally, the inverse kinematics approach is significantly more computationally efficient.

Nevertheless, there are scenarios in which inverse kinematics fails to produce valid angle estimates, for example, when the target position lies beyond the robot's physical reach, defined by the sum of the shoulder-to-elbow ($D_{s\text{-}e}$) and elbow-to-hand ($D_{e\text{-}h}$) segment lengths. In such cases, the exhaustive search method provided by forward kinematics becomes necessary.

Although both the VR application and firmware are implemented as embedded software, our experiments demonstrated that these kinematic computations are best performed on the VR headset due to its substantially greater processing power. Meanwhile, the ESP32-based firmware handles lower-level tasks such as sensor data acquisition, motor control, and communication, ensuring an efficient division of responsibilities across the system.

To assess the impact of miscalibrating the segment lengths $D_{s\text{-}e}$ and $D_{e\text{-}h}$, Figure 6 shows the resulting position errors for several configurations listed in Table I. The analysis assumes the correct reference values to be $D_{s\text{-}e} = D_{e\text{-}h} = 0.5$ m. The plot reveals that multiple joint configurations yield identical error profiles. Specifically, the position errors for points 1, 2, 3, 4, and 7 overlap entirely. For these cases, the

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

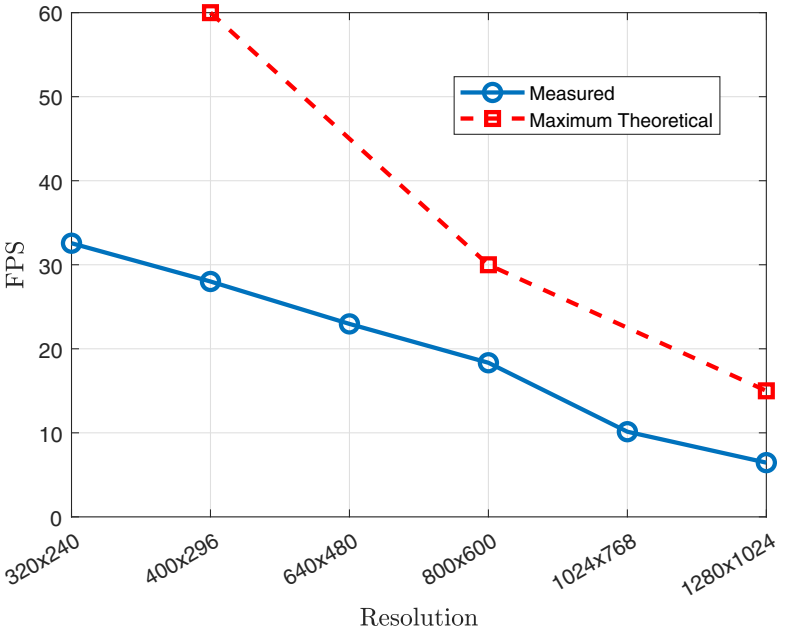


Figure 7. *Average frame rate (FPS) for different image resolutions. The solid line represents the measured average FPS across resolutions, while the dashed red line indicates maximum theoretical FPS values for selected resolutions (400 × 296, 800 × 600, 1280 × 1024).*

error starts at 0.6 m when $D_{s\text{-}e} = D_{e\text{-}h} = 0.2$, decreases linearly to zero at the correct calibration point ($D_{s\text{-}e} = D_{e\text{-}h} = 0.5$), and then increases linearly back to 0.6 m as the values rise to 0.8.

In contrast, points 5 and 6 exhibit a similar trend but with a reduced maximum error of approximately 0.43 m at both ends of the tested calibration range. This behavior highlights the sensitivity of kinematic calculations to accurate measurements of limb segment lengths. Consequently, precise calibration of the shoulder-to-hand distance is essential to ensure reliable position estimation and overall system accuracy.

3.2. FPS and image quality

The proposed system employs an OV2640 camera module to capture image data. According to the manufacturer's specifications, the camera supports a maximum resolution of 1600 × 1200 pixels at 15 frames per second (fps). For lower resolutions, it is capable of achieving 30 fps at 800 × 600 and up to 60 fps at 400 × 296 [38]. However, as illustrated in Figure 7, empirical results reveal that actual performance when integrated with the ESP32 microcontroller falls significantly short of these theoretical limits.

Specifically, at a resolution of 400 × 296, the camera achieved a frame rate of 28 fps – approximately 46.7% of the datasheet's stated maximum. For 800 × 600, we observed a frame rate of 18.3 fps (61% of the theoretical 30 fps), and for 1280 × 1024, 6.45 fps was achieved – just 43% of the expected 15 fps. These discrepancies are primarily attributed to the ESP32's limited processing power and the communication bottlenecks involved in transmitting image data to the VR headset.

Despite these constraints, empirical testing indicates that a resolution of 800×600 provides the most favorable trade-off between image clarity and frame rate for the intended application. While higher resolutions yield better image quality, they come at the cost of a significantly reduced frame rate, negatively impacting real-time performance and user experience.

3.3. Sound, data acquisition, and real-time response

The microcontroller is responsible for managing several time-sensitive tasks, including real-time image acquisition and transmission, audio sampling via its ADC, motor control through its timer peripherals,

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

and audio playback using the RMT (Remote Control) peripheral. Experimental results revealed that to minimize high-frequency noise and enhance audio clarity, each sample of the 8 kHz audio signal should be repeated at least three times during playback. Although higher repetition rates were evaluated, they significantly increased the computational load on the already resource-constrained system without yielding perceptible improvements in sound quality. Among the peripherals tested for audio output, the RMT module consistently produced the most reliable and clear results.

Regarding communication and system responsiveness, TCP was found to offer greater reliability compared to UDP. Although UDP offers lower latency, its lack of guaranteed delivery introduced issues such as missed motion commands and intermittent audio dropouts. These problems were particularly noticeable during fast operator inputs and simultaneous audio transmission. Consequently, TCP was adopted for its superior stability in the context of this application.

Finally, while image data is continuously streamed from the Jaeger UTFPR system, experimental evaluation demonstrated that a positional update rate of 6 Hz from the VR headset strikes the optimal balance between responsiveness and system load, leading to a more consistent user experience.

3.4. Key challenges

Designing Jaeger UTFPR under strict cost constraints introduced a series of technical and engineering challenges that required innovative solutions across hardware, firmware, and software layers. Every design decision – from mechanical structure to embedded system architecture – was guided by the goal of achieving functionality, reliability, and interactivity at an exceptionally low price point.

A core challenge lay in selecting components that could collectively deliver real-time control, audio-visual streaming, and responsive actuation with low cost. The ESP32 microcontroller was chosen for its rare balance of affordability, wireless capabilities, and sufficient processing power to simultaneously handle tasks such as image acquisition, stepper motor control, and audio signal processing. However, maximizing the utility of such constrained hardware demanded optimized firmware design, careful peripheral usage, and strict memory and timing management.

Image capture relied on the OV2640 camera module, which supports resolutions up to UXGA while offering low latency at modest frame rates. Transmitting visual data over Wi-Fi in parallel with real-time audio and motor commands introduced substantial synchronization and bandwidth management challenges. Given the ESP32's limited resources, the firmware was meticulously optimized to prioritize tasks based on timing sensitivity, buffering strategies, and transmission modes.

The actuation system posed its own set of complexities. A hybrid approach using low-cost N20 DC motors and small stepper motors was adopted to deliver adequate degrees of freedom while maintaining positional accuracy. However, controlling nine motors with limited GPIOs necessitated the use of 74HCT259 addressable latches in combination with DRV8833 motor drivers. Implementing this multiplexed control scheme required precise logic sequencing and signal routing to ensure reliable operation without interference – particularly under concurrent Wi-Fi communication and sensor processing.

Audio processing added further constraints. Real-time audio transmission demanded careful tuning of sampling rates, quantization schemes, and buffering techniques. To reduce ESP32 workload while maintaining intelligibility, audio input was sampled at 24 kHz, downsampled to 8 kHz, and compressed into 8-bit values before being transmitted in 500 ms chunks. Playback using the RMT peripheral required creative signal generation techniques to synthesize PWM waveforms with minimal CPU overhead.

On the software side, the VR application was tasked with offloading computationally intensive processes, such as inverse kinematics and hand tracking, from the microcontroller. Ensuring accurate, low-latency coordination between the VR interface and the physical robot required careful calibration and networking strategies, as well as support for both TCP and UDP protocols for balancing reliability and real-time responsiveness.

For the motors, a combination of N20 DC motors and stepper motors was selected to ensure the necessary degrees of freedom and movement precision without incurring unnecessary costs. Similarly, using readily available and low-cost components, such as the DRV8833 motor drivers and LM358 operational amplifiers, ensures that the overall system remains affordable while meeting the required performance

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

Table II. *Estimated costs, with a total of less than USD 35 [39–41].*

Item	Price (USD)	Quantity	Total (USD)
ESP32 Module	2.50	1	2.50
Camera	2.00	1	2.00
DC N20 Motor	0.80	2	1.60
Stepper Motor	1.50	7	10.50
Other ICs and Items	7.00	1	7.00
3D Printed Parts	4.00	1	4.00
PCB Fabrication	3.50	2	7.00

specifications. This focus on cost-effective solutions allows the robot to deliver functional capabilities while minimizing the financial burden of production. The total cost of producing a single robot is estimated at below USD 35 [39–41], as summarized in Table II. The detailed electronics bill of materials (BOM), including "Other ICs and Items," is available in the full BOM listed in ref. [35].

While the unit cost is low, there are a few important considerations to keep in mind. First, it does not account for shipping costs or taxes. Since many parts need to be sourced from different vendors, shipping costs can easily exceed the price of the individual components. Additionally, although the cost of PCB fabrication is included, producing a single PCB is significantly more expensive than manufacturing multiple units. To mitigate these issues, one should purchase as many components as possible from each supplier. Furthermore, producing more than one unit at a time is considerably more cost-effective than manufacturing a single unit.

For the 3D-printed parts, which are the primary structural components, all necessary models are publicly available in ref. [35] as 16 .stl files, totaling approximately 20 individual parts. The total filament consumption is about 80g. In the project's experiments, we used PLA filament, but other materials could be used as well. The material cost for a single robot is less than $10. We also recognize that for an individual user, the cost of the 3D printer itself can be a hidden threshold. However, given the open-source nature of the project, users can also choose to have the parts printed by a third-party service, and the full BOM is also available in ref. [35]. Note that the unit cost of ICs and other components could be significantly reduced for mass production [39–41]. It is also worth noting that the maintenance cost of the printer (wear and tear, electricity, etc.) represents a negligible cost of a few cents for this project, making the overall cost of the 3D printed parts highly affordable. Moreover, the open-source nature of the project encourages users to share their own designs and improvements, further reducing the overall cost of the project.

In summary, the development of Jaeger UTFPR demanded far more than assembling inexpensive components – it required creative engineering to overcome the limitations inherent in ultra-low-cost design. The result is a fully functional, immersive telepresence robot system built on embedded hardware that demonstrates how performance and affordability can coexist through thoughtful system integration.

3.5. Applicability

As a remote-controlled small robot, the Jaeger UTFPR offers a wide range of applications. In education, it provides an engaging way to introduce fundamental robotics concepts, fostering problem-solving skills and creativity. The robot's small size is a key advantage for its application in education, making it an ideal tool for teaching fundamental robotics concepts in a hands-on, classroom setting. Its compact design allows for easy storage and deployment, and its open-source nature provides students with a complete platform to learn about hardware design, firmware development, and VR programing. The same characteristics also make it suitable for educational competitions, where navigation in tight spaces is often required. Its low cost makes it accessible to a wide range of educational institutions and students, easing access to robotics education.

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

With a full arm length of just 7 cm, the robot's arms are designed for simple tasks, such as interacting with small objects or gesturing, rather than high-precision manipulation. The stepper motors offer sufficient resolution for educational and entertainment applications, such as interacting with small objects or performing simple gestures. However, for applications such as foreign object detection in pipelines, its primary capability comes from the combination of its small size, maneuverability, and video feedback, which allows the operator to perform visual inspections in confined spaces. With further improvements, the robot could serve in exploration roles in difficult-to-reach areas, such as construction, maintenance, and disaster recovery, or be expanded to fields like surveillance and remote monitoring.

The Jaeger UTFPR fills a critical gap when compared to other market solutions. While many miniature remote-controlled robots exist, there are very few affordable VR-controlled platforms. For complex, bipedal robots, prices can range into the thousands of dollars, such as with the Unitree R1 EDU [42]. Even more accessible options, like the InMoov DIY project [43], can have a cost that is at least 10 times higher than the one presented in this work. This platform therefore provides a fully open-source, VR-controlled humanoid at an unprecedentedly low cost, contrasting with commercial solutions that offer superior performance at a significantly higher price and often lack an open-source nature. The primary focus of this platform is on visual inspection and remote presence, rather than on manipulation precision, which would require more expensive actuators and encoders.

3.6. Limitations and future work

We acknowledge that the design of the Jaeger UTFPR under strict cost constraints introduces several limitations that are important to discuss. First, the multiplexing scheme for motor control using 74HCT259 latches and DRV8833 drivers, while cost-effective, can introduce a slight latency in motor response. This is because the controller must address each motor sequentially, which can be a bottleneck in scenarios requiring simultaneous, high-speed movements of multiple stepper motors. Future work could explore the use of dedicated, multi-channel motor driver ICs to reduce this latency and improve the real-time responsiveness of the system.

Second, while the 800×600 resolution provides a reasonable trade-off between image quality and frame rate, a frame rate of 18.3 fps and observed latencies of 20–60 ms may not be sufficient for highly dynamic teleoperation tasks, such as navigating an environment with rapid changes in scenery. Optimizing the image compression algorithm on the ESP32 or offloading this task to a more powerful microcontroller could help increase the frame rate and reduce latency without compromising image quality. We also note that the system currently does not have a specific alternative for a complete lack of a Wi-Fi signal, which is a key limitation for teleoperation in unstable network conditions. Alternatives to Wi-Fi may include Narrowband-Ultra-Reliable Low-Latency Communication. It could be an option if more reliability is required in a new version of the robot, but it would increase the costs and complexity of the network infrastructure.

The system's power consumption varies significantly with its state of operation. The main board, responsible for streaming and processing, consistently draws about 300 mA. However, the total current draw can increase to approximately 900 mA when the robot's arms and wheels are actively moving, due to the high consumption of the stepper motors (about 500 mA each) and DC motors (about 50 mA each). The two 1300 mAh batteries, connected in series, allow for an average operating time of approximately 1 to 2 hours, taking into account the inherent power loss in the LDO voltage regulators. This endurance can be extended in scenarios with minimal movement. For prolonged use, a larger battery pack or an external power source would be required. This is a crucial limitation for applications such as remote inspection, where extended operation is often necessary. Future work could focus on optimizing power consumption and exploring more efficient battery solutions to increase the robot's endurance.

4. Conclusion

This paper presents the design and implementation of Jaeger UTFPR, an open-source, low-cost, remote-controlled tiny humanoid robot measuring just 12 cm in height. Developed with a focus on accessibility

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

and affordability, the robot integrates 3D-printed components, cost-effective electronics, and wireless communication to deliver real-time audio and video feedback through a VR interface. The embedded system, consisting of the VR application and the ESP32 firmware, enables control and communication between the operator and the robot. Operators control Jaeger UTFPR via a VR headset and motion controllers, allowing for immersive telepresence and intuitive manipulation of the robot's movements. Experimental evaluations confirm the system effectively balances cost and performance, showcasing its potential for immersive robotics applications across education, research, entertainment, and remote inspection.

By using additive manufacturing, modular electronics, and a VR-based control interface, Jaeger UTFPR demonstrates that meaningful humanoid robot interactions can be achieved even at miniature scales and minimal budgets. Unlike other platforms that require costly actuators or extensive proprietary hardware, this work emphasizes simplicity, replicability, and open access. The results suggest a valuable paradigm for democratizing robotics research, lowering the barriers for students, hobbyists, and educational institutions alike. Additionally, the immersive experience of inhabiting a 12 cm-tall perspective offers novel human–robot interaction opportunities not typically explored in conventional humanoid robotics.

Author contributions. Marcos Eduardo Pivaro Monteiro: Conceptualization; Investigation; Methodology; Writing – Original Draft; Preparation. Eduardo Nunes dos Santos: Writing – Review and Editing; validation. Daniel Fernando Pigatto: Formal Analysis; Writing – Review and Editing. Joilson Alves Junior: Validation; visualization. Leonardo Lira Ramalho: Writing – Review and Editing; Visualization. Diego Issicaba: Formal Analysis; Validation.

Financial support. This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] G. McCartney and A. McCartney, "Rise of the machines: Towards a conceptual service-robot research framework for the hospitality and tourism industry," *Int. J. Contemp. Hosp. Manag.* **32**(12), 3835–3851 (2020).
[2] F. McLeay, V. S. Osburg, V. Yoganathan and A. Patterson, "Replaced by a robot: Service implications in the age of the machine," *J. Serv. Res.* **24**(1), 104–121 (2021).
[3] V. Yoganathan, V.-S. Osburg, W. H. Kunz and W. Toporowski, "Check-in at the robo-desk: Effects of automated social presence on social cognition and service implications," *Tour. Manag.* **85**, 104309 (2021).
[4] A. Asghar, A. Patra and K. S. Ravi, "The potential scope of a humanoid robot in anatomy education: A review of a unique proposal," *Surg. Radiol. Anat.* **44**(10), 1309–1317 (2022).
[5] J. J. Steil, F. Röthling, R. Haschke and H. Ritter, "Situated robot learning for multi-modal instruction and imitation of grasping," *Robot. Auton. Syst.* **47**(2-3), 129–141 (2004).
[6] G. Taga, "Global entrainment in the brain–body–environment: Retrospective and prospective views," *Biol. Cybern.* **115**(5), 431–438 (2021).
[7] T. Reil and P. Husbands, "Evolution of central pattern generators for bipedal walking in a real-time physics environment," *IEEE Trans. Evol. Comput.* **6**(2), 159–168 (2002).
[8] T. Sugihara and M. Morisawa, "A survey: Dynamics of humanoid robots," *Adv. Robot.* **34**(21-22), 1338–1352 (2020).
[9] Y. Tong, H. Liu and Z. Zhang, "Advancements in humanoid robots: A comprehensive review and future prospects," *IEEE/CAA J. Autom. Sin.* **11**(2), 301–328 (2024).
[10] P. T. Zacharia and E. K. Xidias, "Optimal robot task scheduling in cluttered environments considering mechanical advantage," *Robotica* **42**(9), 3230–3246 (2024).
[11] M. S. Khan and R. K. Mandava, "A review on gait generation of the biped robot on various terrains," *Robotica* **41**(6), 1888–1930 (2023).
[12] H. Xia, Z. Li, G. Chen, H. Qiao and J. S. Dai, "Intelligence in robotics for computer, engineering, and applied sciences," *Robotica* **42**(7), 2085–2088 (2024).
[13] Z. Lv, Y. Wang, G. Zhang and X. Wang, "Determination method of limit interpolation points in industrial robot control system," *Robotica* **43**(4), 1533–1547 (2025).
[14] K. A. Szczurek, R. M. Prades, E. Matheson, J. Rodriguez-Nogueira and M. D. Castro, "Multimodal multi-user mixed reality human–robot interface for remote operations in hazardous environments," *IEEE Access* **11**, 17305–17333 (2023).

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press

[15] M. Goodrich and A. Schultz, "Human–Robot Interaction: A Survey," **In:** *Foundations and Trends in Human-Computer Interaction*, Now Publishers Inc., vol. 1 (2007) pp. 203–275.
[16] M. Gallipoli, S. Buonocore, M. Selvaggio, G. A. Fontanelli, S. Grazioso and G. Di Gironimo, "A virtual reality-based dual-mode robot teleoperation architecture," *Robotica* **42**(6), 1935–1958 (2024).
[17] M. J. Sobrepera, A. T. Nguyen, E. S. Gavin and M. J. Johnson, "Insights into the deployment of a social robot-augmented telepresence robot in an elder care clinic – perspectives from patients and therapists: A pilot study," *Robotica* **42**(5), 1321–1349 (2024).
[18] S. Tachi, *Telexistence* , 2nd ed. (World Scientific Publishing Company, Singapore, 2015).
[19] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi and D. Pucci, "Teleoperation of humanoid robots: A survey," *IEEE Trans. Robot.* **39**(3), 1706–1727 (2023).
[20] M. Fujita, Y. Kuroki, T. Ishida and T, "A Small Humanoid Robot SDR-4X for Entertainment Applications," **In:** *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, IEEE, vol. 2 (2003) pp. 938–943.
[21] M. Lapeyre, P. Rouanet and P.-Y. Oudeyer, "The Poppy Humanoid Robot: Leg Design for Biped Locomotion," **In:** *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE (2013) pp. 349–356.
[22] M. Lapeyre, S. N'Guyen, A. L. Falher and P.-Y. Oudeyer, "Rapid morphological exploration with the poppy humanoid platform," **In:** 2014 IEEE-RAS International Conference on Humanoid Robots, IEEE, (2014) pp. 155–160.
[23] A. J. Fuge, C. W. Herron, B. C. Beiter, B. Kalita and A. Leonessa, "Design, development, and analysis of the lower body of next-generation 3D-printed humanoid research platform: PANDORA," *Robotica* **41**(7), 2177–2206 (2023).
[24] J. Y. Kim, Z. Kashino, T. Colaco, G. Nejat and B. Benhabib, "Design and implementation of a millirobot for swarm studies – *mROBerTO*," *Robotica* **36**(11), 1591–1612 (2018).
[25] D. T. de Paula, E. P. Godoy and M. Becerra-Vargas, "Dynamic modeling and simulation of a torque-controlled spatial quadruped robot," *Robotica* **42**(8), 2761–2780 (2024).
[26] A. Müller, M. Brandstötter, L. Romdhane and A. Gasparetto, "Editorial to the special issue on "Modeling and control of innovative robots"," *Robotica* **42**(6), 1710–1711 (2024).
[27] J. Li, C. Liu, K. Nguyen and J. M. McCarthy, "A steerable robot walker driven by two actuators," *Robotica* **42**(12), 4019–4035 (2024).
[28] L. Liang, Y. Xu, L. Han and Y. Liu, "System design and control of the sphere-wheel-legged robot," *Robotica* **42**(10), 3363–3379 (2024).
[29] M. A. Arif, A. Zhu, H. Mao and Y. Tu, "Panoramic visual system for spherical mobile robots," *Robotica* **42**(7), 2089–2107 (2024).
[30] O. Selvi, C. Cindioglu, S. K. Ider and M. Ceccarelli, "Kinematic design and analysis of a novel overconstraint walking mechanism," *Robotica* **43**(4), 1–21 (2025).
[31] T. Fukuda, H. Dobashi, H. Nagano, Y. Tazaki, R. Katayama and Y. Yokokohji, "Jigless assembly of an industrial product by a universal robotic hand mounted on an industrial robot," *Robotica* **41**(8), 2464–2488 (2023).
[32] M. Ojer, A. Etxezarreta, G. Kortaberria, B. Ahmed, J. Flores, J. Hernandez, E. Lazkano and X. Lin, "High accuracy hybrid kinematic modeling for serial robotic manipulators," *Robotica* **42**(9), 3211–3229 (2024).
[33] L. Almeida, V. Santos and J. Ferreira, "Enhancement of humanoid robot locomotion on slippery floors using an adaptive controller," *Robotica* **42**(4), 1055–1073 (2024).
[34] J. Li, D. Cong, Y. Yang and Z. Yang, "A hydraulic actuator for joint robots with higher torque to weight ratio," *Robotica* **41**(2), 756–774 (2023).
[35] M. E. P. Monteiro, "Jaeger UTFPR Repository: Source code for robotic control" (2024). Accessed 28 May 2025. Available at: https://github.com/TechnologicalCraftsmanship/JaegerUTFPR.
[36] A. Payne, "Downsizing (Motion Picture)" (2017). Accessed 24 November 2025. Available at: https://www.imdb.com/title/tt1389072/.
[37] I. Raptis and B. Sullivan, "Arachne system: An accessible miniaturized mobile multi-robot platform," *Robotica* **43**(7), 2610–2643 (2025).
[38] OmniVision Technologies Inc., "OV2640 Color CMOS UXGA (2.0 MegaPixel) CameraChip Datasheet (v. 1.6)" (2006). Accessed 28 May 2025. Available at: https://www.datasheetbank.com/preview/OV2640-OMNIVISION-V2.
[39] Alibaba Group, "Alibaba E-commerce Platform: Global Trade Marketplace" (2025). Accessed 28 May 2025. Available at: https://www.alibaba.com/.
[40] Mouser Electronics, "Electronic Components Distributor Catalog" (2025). Accessed 28 May 2025. Available at: https://www.mouser.com/.
[41] PCBWay, "PCBWay - PCB prototype" (2025). Accessed 28 May 2025. Available at: https://www.pcbway.com/.
[42] Unitree Robotics, "Unitree R1: Humanoid Robot Platform Specifications" (2024). Accessed 23 August 2024. Available at: https://www.unitree.com/R1.
[43] G. Langevin, "InMoov: Open Source 3D Printed Life-Size Robot" (2012). Accessed 23 August 2024. Available at: https://inmoov.fr/.

Cite this article: M. E. Pivaro Monteiro, E. N. dos Santos, D. F. Pigatto, J. Alves Junior, L. L. Ramalho and D. Issicaba (2025). "Design and implementation of an open-source, low-cost remote-controlled tiny humanoid robot for accessible robotics applications", Robotica **43**, 4366–4383. https://doi.org/10.1017/S0263574725102907

https://doi.org/10.1017/S0263574725102907 Published online by Cambridge University Press