International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Development and User Testing of Gesture-Controlled Robotic Arms: A Secondary Study

Rashvin Vij

Abstract
Purpose: This secondary study analyzes the development of gesture-controlled robotic arms and how their usability has been evaluated. We review recent advances in Human–Robot Interaction (HRI) that enable users to command robotic arms through hand and arm gestures. Method: A literature-based approach was used, surveying peer-reviewed studies from 2015–2025 on gesture recognition techniques and user testing protocols. Key insights: We found that gesture control is implemented via wearable sensors (e.g. flex sensors, electromyography) and vision-based systems (e.g. depth cameras, Leap Motion), often enhanced by artificial intelligence for real-time recognition. Usability evaluations commonly involve task performance metrics and user feedback, revealing trends such as increasing accuracy (often >90%), the integration of machine learning for more natural control, and recognition of user-centered design as critical. Conclusion: Gesture-controlled robotic arms can provide intuitive, human-centric control in domains from industry to assistive technology. However, rigorous user testing with diverse participants in realistic scenarios is needed to ensure these systems are safe, effective, and truly user-friendly. Embracing user-centered design and iterative evaluation will be vital for deploying gesture-controlled robots in real-world applications.

1. Introduction

Gesture-controlled robotic arms are robotic manipulators that can be directed through human gestures – such as hand motions or poses – rather than traditional interfaces like joysticks or keyboards. This mode of control is a key aspect of Human–Robot Interaction (HRI), a field focusing on natural and intuitive communication between humans and robots. By interpreting movements of a user's hand or arm, a robot can mirror those movements or execute corresponding actions, creating a more seamless collaboration between human intent and machine action. Such gesture-based control is inherently human-centric, aligning with the Industry 5.0 vision of intuitive and "zero-training" interfaces for robotics.

Application Areas: Gesture-controlled arms have broad applications across industries and assistive domains. In healthcare, surgeons have explored using gestures to tele-operate surgical robotic arms, allowing sterile, touch-free control of instruments. Rehabilitation and prosthetics also benefit: intuitive movement-based prosthesis control enables amputees to coordinate artificial arms using natural motions or muscle signals, improving their daily function. In industrial and manufacturing settings, gesture-controlled robotic arms can facilitate safer collaboration on assembly lines – for example, a worker's hand movements can guide a collaborative robot to lift or position parts without direct contact. In education and research, such systems are used as interactive tools to teach robotics, allowing students to control robot arms via simple hand motions. Additionally, assistive technology stands to gain significantly: individuals with mobility impairments or older adults can use gestures (including residual limb movements or simple hand poses) to control assistive robotic arms for daily tasks, such as feeding or object retrieval, thereby enhancing independence . Overall, the ability to command robots through gestures opens possibilities in any scenario where touchless, intuitive control is advantageous – from controlling service robots in smart homes to directing robotic arms in hazardous environments (e.g. bomb disposal or high-temperature manufacturing) where keeping humans at a safe distance is crucial.

Research Purpose and Scope: This paper aims to survey how gesture-controlled robotic arms have been developed and evaluated over the past decade. We focus on two intertwined aspects: (1) the technologies and design of gesture recognition and robotic arm systems, and (2) the user testing and human-centered evaluation of these systems. By reviewing recent (2019–2025) literature, we identify the prevailing gesture recognition methods (sensor-based, vision-based, and hybrid AI approaches) and the hardware/software integration of robotic arms, including mechanical design and control architecture. Simultaneously, we analyze how researchers have assessed usability – e.g. in lab tasks or field trials – and what those evaluations reveal about system performance, user satisfaction, and remaining challenges. The scope is deliberately broad, covering applications from industrial robotics to prosthetic arms, to extract common findings and gaps. Ultimately, this secondary study is intended to highlight best practices in developing gesture-controlled robotic arms and underscore the importance of user-centered testing in advancing these technologies.

2. Methodology of Secondary Research

We conducted a secondary literature review to gather and synthesize existing research on gesture-controlled robotic arms. The review process involved searching multiple academic databases and digital libraries, including IEEE Xplore, ACM Digital Library, ScienceDirect (Elsevier), SpringerLink, and MDPI, for articles published between 2015 and 2025. We also included relevant open-access journals and high-quality conference proceedings to ensure comprehensive coverage of recent developments. Key search terms combined concepts of gesture control or hand gesture recognition with robotic arms, robotic manipulators, prosthetics, and usability or user study.

Inclusion criteria: We filtered for peer-reviewed studies (journal papers, conference papers, and a few authoritative review articles) that met the following conditions: (a) they involve a robotic arm or manipulator controlled via human hand/arm gestures or motions; (b) they discuss the development of the gesture recognition system or the robotic arm hardware/software; and/or (c) they evaluate the system with human users or through experiments measuring usability, performance, or accuracy. We prioritized literature from 2019 onwards to capture the state-of-the-art, though earlier seminal works were considered for background. After initial filtering, dozens of relevant sources were identified, from which we extracted technical details and evaluation results.

Analysis approach: We performed a thematic analysis of the selected literature. This entailed coding each source for themes such as the gesture recognition technique used (e.g. wearable sensor, vision-based, machine learning algorithm), the robotic arm setup (hardware design, degrees of freedom, control platform), and the user testing protocol (number of participants, task scenarios, metrics recorded). We then grouped findings into coherent themes corresponding to our sections: Gesture Recognition Technologies (comparing sensor-based vs. vision-based vs. hybrid approaches), Robotic Arm Development (mechanical and control integration considerations), User Testing and Human-Centered Evaluation (methods and outcomes of user studies), and so on. Within each theme, we compared results across studies to identify trends, strengths, and gaps. For example, we compared accuracy rates of different gesture recognition methods, or the participant feedback from studies using gloves versus camera-based control. The insights from these comparisons were used to inform the comparative analysis (Section VII) and to suggest future directions. All information is drawn from the reviewed sources and is cited accordingly to maintain academic rigor. Being a secondary study, no new experiments were conducted; instead, we rely on reported data and observations, cross-verifying findings across multiple studies where possible.

3. Gesture Recognition Technologies: A Literature Review

Gesture recognition is the cornerstone of any gesture-controlled robotic arm system. We summarize three main categories of techniques for capturing and interpreting human gestures: sensor-based methods, vision-based methods, and AI-powered or hybrid approaches. Each approach has distinct advantages and limitations, which we discuss below.

3.1 Sensor-Based Methods

Figure 1: Example of a sensor-based gesture control setup – a user wears a custom data glove with flex sensors (for finger bending detection) and an inertial measurement unit, enabling intuitive control of a robotic arm.

Sensor-based gesture recognition relies on wearable or handheld devices that directly measure the user's movements or muscle activity. A common example is the data glove: a glove fitted with flex sensors (bend sensors) on the fingers to detect finger curl, often combined with inertial measurement units (IMUs) like accelerometers/gyroscopes to track hand orientation. When a user moves their hand or bends their fingers, the sensors produce signals that can be mapped to robot commands. For instance, bending the index finger could signal the robot to close its gripper, while moving one's hand up moves the robotic arm up. These wearable systems have the benefit of low latency and independence from environmental conditions (lighting or line-of-sight are not issues), since they directly measure the user's body. They can be very precise in measuring finger angles – the glove in one study captured finger positions with an average error of only ~1 cm when replicated by the robot. Sensor gloves also tend to be affordable and DIY-friendly; for example, simple flex sensor gloves with an Arduino microcontroller can cost on the order of \$200, making them accessible for education and prototyping.

Another sensor-based approach uses electromyography (EMG), which involves reading the electrical activity of muscles. Surface EMG electrodes placed on the forearm can detect muscle signals when the user intends to move their hand or fingers. These signals can be interpreted as specific gestures or grip patterns to control a prosthetic hand or robot arm. EMG-based control is popular in prosthetic limb research, since it enables an amputee to operate a robotic hand by contracting the same muscles that would move a biological hand. With recent advances in deep learning, EMG pattern recognition accuracy has greatly improved (studies report >90% accuracy distinguishing multiple hand gestures via EMG signals). The appeal of EMG is that it can provide intent even if no actual movement occurs (useful when the physical hand is absent or fixed).

Despite their promise, sensor-based methods come with challenges. Wearability and comfort can be issues – users must don equipment like gloves or armbands, which can be cumbersome or constrain natural motion if not designed ergonomically. IMU sensors, while mobile, can suffer from drift over time, leading to decreasing accuracy without frequent recalibration. An IMU-based glove may gradually lose track of absolute orientation, causing the robot's perceived hand position to "wander" – one reviewed paper noted that drift errors and complex calibration procedures were a key drawback of purely inertial gloves. Similarly, EMG control requires calibration for each user and can be sensitive to electrode placement and muscle fatigue, introducing variability. Noise in sensor signals is another concern – for instance, flex sensors can be noisy and have non-linear response curves , which necessitates filtering or calibration for precise control.

Overall, sensor-based gesture recognition provides a direct, fast link from human motion to robot action and works reliably in varied environments, but at the cost of requiring users to wear devices and manage issues like drift and calibration. Many systems mitigate these issues by combining multiple sensors (e.g. adding a magnetometer to an IMU glove to correct drift, or using hybrid sensor suites), which leads to the hybrid approaches discussed in section 4.3.

3.2 Vision-Based Methods

Vision-based gesture recognition uses cameras or optical sensors to observe the user's movements without requiring any wearables. A prominent example is the Leap Motion controller, a small infrared stereo camera device placed on a desk or VR headset, which can track hand and finger positions in mid-air. Vision-based methods also include standard RGB cameras or RGB-D depth cameras (like Microsoft Kinect or Intel RealSense) that capture the user's body pose in three dimensions. These systems seek to perform markerless tracking, recognizing the user's gestures from video input alone.

The appeal of vision-based approaches is that they allow truly contactless interaction – the user can gesture freely without being tethered or instrumented. This can feel more natural and less intrusive. For instance, in industrial settings a worker might simply wave a hand or perform a predefined gesture to command a robot from a distance. In one study, a Leap Motion was used for a collaborative robot arm: the user's left hand position controlled the arm's end-point and the right hand's opening/closing controlled the gripper. This setup yielded real-time control of a 6-DOF manipulator, and a motorized stage even adjusted the sensor's height to fit different users – demonstrating how vision-based systems can be made adaptive and convenient. Another study employed a Kinect depth sensor to track a person's upper body skeleton; by analyzing the 3D joint data, the system identified the user's left-hand gesture as a command and simultaneously mapped the right arm's motions to a robot arm's joints via inverse kinematics. This two-hand Kinect-based interface allowed both discrete commands and continuous teleoperation of the robot in real time.

However, vision-based gesture recognition faces its own set of challenges. These systems are highly dependent on lighting, background, and sensor placement. For example, the Leap Motion has an effective range of about 60–80 cm from the device and a limited field of view (approximately 140° by 120°) – if the user's hands move outside this volume or occlude each other, tracking is lost. Similarly, depth cameras like the Kinect require a clear line-of-sight; any occlusion (e.g. one arm blocking the other, or poor lighting for an RGB camera) can degrade performance. The precision of vision tracking can be an issue for fine motions: one report noted that depth sensors struggle with detecting very small movements (like slight finger pinches), causing erratic distance readings. Vision systems also impose a computational load – the raw image or depth data must be processed by computer vision algorithms or neural networks in real time, which may introduce latency if not optimized.

Nonetheless, recent progress in camera-based hand tracking has improved robustness. Modern methods leverage convolutional neural networks to detect hand keypoints even under difficult visual conditions. Depth cameras provide direct 3D data that, combined with AI, can yield accurate hand pose estimation. For example, one system used an RGB-D camera with a pose estimation model to get 21 hand joint positions in real time. Such rich visual information can enable very natural mapping – the reviewed assembly-line study translated the user's observed hand pose into a 7-DOF robot arm motion with an intuitive one-to-one correspondence. Vision-based methods also shine in full-body or large-workspace scenarios where wearable solutions would be impractical; a single depth sensor can capture arm, torso, and even full-body gestures to control remote robots or drones.

In summary, vision-based gesture control offers a wearable-free experience and can leverage the expressive power of visual communication (think of controlling a robot with a simple wave or pointing gesture). The trade-off lies in dealing with environmental constraints and ensuring the sensing is robust. Many systems now combine vision with AI algorithms to improve reliability, which leads us to hybrid approaches.

3.3 AI & Hybrid Approaches

Artificial intelligence has increasingly been integrated into gesture recognition to boost accuracy and flexibility. AI-driven approaches typically involve training machine learning models (often deep neural networks) on gesture data – whether sensor signals or images – so the system can automatically classify or regress to the correct command. For example, researchers have trained convolutional neural networks (CNNs) to recognize static hand poses from camera images, and recurrent networks (LSTM or Bi-LSTM) to interpret dynamic gesture sequences or streaming sensor data. These AI models can capture complex patterns that rule-based algorithms might miss, improving recognition rates especially when gestures vary between users.

One notable trend is the use of pre-trained hand tracking models like MediaPipe (by Google) or specialized pose estimation networks. MediaPipe provides a real-time 3D hand skeleton from a single RGB camera using deep learning. This effectively gives 21 keypoints of the hand (fingertips, joints, etc.) that can be used to infer gestures. Several recent robotic arm interfaces leverage such AI-generated keypoints instead of raw sensor inputs. In an assembly-line control study, the system identified the operator’s hand joints via an AI model (trained on large hand image datasets) and achieved precise control by mapping those joint coordinates to the robot arm’s target angles. The use of AI here eliminated the need for the user to wear trackers on each joint – the software learned to “see” the hand and interpret its orientation and gestures reliably. Indeed, deep learning-based vision has become mainstream in gesture recognition, gradually replacing earlier heuristic or hardware-based methods. The data richness and generalization ability of neural networks allow systems to cope with variations in hand shape, size, or even lighting to some extent.

Hybrid systems combine multiple modalities – for instance, merging sensor-based and vision-based data – to capitalize on the strengths of each. An illustrative example is the agricultural harvesting robot arm controller, which used a hybrid of a bending-sensor glove and an OptiTrack motion capture system. The glove provided detailed finger bend data (to recognize discrete gestures like pinch or fist), while the OptiTrack cameras provided highly accurate 3D position tracking of the hand via reflective markers. A custom CNN+BiLSTM model fused these inputs to classify gestures with 96% accuracy and replicated the hand’s position with only ~1 cm error. This fusion mitigated individual drawbacks: the motion capture fixed the drift problem of IMUs, and the glove data filled in finger pose details that pure vision might miss. Similarly, other research has fused EMG signals with inertial data, or depth camera data with IMU, allowing the system to remain robust if one signal source is noisy or lost. The general insight is that sensor fusion combined with AI can yield a more resilient and accurate gesture recognition than any single method alone.

AI also enables adaptive and personalized gesture control. Machine learning models can be trained on an individual’s gesture style, allowing the system to adjust to different users (addressing the inconsistency where developers’ assumed gestures might differ from what users naturally do). Some studies incorporate feedback loops where AI algorithms learn from the user's corrections or use reinforcement learning to refine the control policy over time. For instance, an AI might predict the user's intended motion and smooth out jitter or fill in gaps, resulting in more stable robot movement – a technique noted as a future direction in improving prosthetic hand control.

In summary, AI-powered approaches have become indispensable in cutting-edge gesture-controlled systems. They greatly improve recognition accuracy and allow creative combinations of inputs. The downside is the need for sufficient training data and computational resources, and the "black box" nature of deep learning which can make the system's errors hard to predict. However, when carefully deployed, AI and hybrid methods significantly enhance the naturalness and reliability of gesture control, bringing us closer to intuitive, user-friendly robotic arms.

4. Robotic Arm Development: Mechanisms and Control

To fully realize a gesture-controlled system, the robotic arm itself and its control architecture must be designed or selected with care. Here we discuss the hardware integration of robotic arms (mechanical components and degrees of freedom), the embedded systems and communication links that connect user inputs to the robot, and design considerations such as cost vs. complexity.

4.1 Hardware Integration

The "arm" in a gesture-controlled robotic arm can range from a simple 3-degree-of-freedom (DOF) pick-and-place manipulator to an anthropomorphic 7-DOF robotic arm. Degrees of freedom correspond to the number of independent joints or axes the arm can move in. A higher DOF arm (like the 7-DOF Franka Emika Panda used in one study) offers greater flexibility – it can reach around obstacles and orient its end-effector in multiple ways to accomplish the same task. This is ideal for mimicking human arm capabilities but also increases mechanical and control complexity. Conversely, lower-DOF arms (3–4 DOF) are simpler, cheaper, and easier to control but less versatile in movement (they might only operate in a plane or have limited orientations). Researchers must choose an appropriate DOF based on the application: for instance, prosthetic arms often prioritize multiple DOFs in the hand for finger motion but may simplify the wrist/elbow, whereas an industrial collaborator robot might use 6 DOF for full spatial positioning plus a gripper.

In terms of construction, many prototypes use servo motors at the joints for precise position control. Hobbyist or low-cost arms often use standard RC servos (which are affordable and easily controlled by microcontrollers), and these arms can be 3D-printed to reduce cost. A 3D printed frame with servos can yield, for example, a 5-axis robot arm for a few hundred dollars – sufficient for demonstrating gesture control concepts. The trade-off is that such arms have limited payload and accuracy. Higher-end arms use brushless DC motors with encoders or stepper motors for better precision and strength, often with gearboxes or belts. In the literature, some teams opted to integrate existing robotic arm kits or commercial robot manipulators: e.g. a 5-DOF robotic arm kit controlled via Arduino, or the Kinova JACO 6-DOF arm used in a lab experiment to test gesture interfaces. If fine manipulation is required, the end-effector is a crucial component – many designs include an adaptive gripper or multi-fingered hand. For example, the Panda arm's rigid gripper (80 mm span) was used in a gesture control experiment, though the specific gripper type did not affect results as long as it could open/close as commanded . Some research prototypes even attach robotic hands with individual finger actuation to mimic human hand gestures more directly.

Kinematic linkages and materials also matter. Lightweight arms (aluminum or 3D-printed plastic) are preferable for wearables or mobile applications (like a shoulder-mounted assistive arm), whereas industrial arms are often steel for rigidity. Backdrivability and safety are concerns if the arm works around humans: in collaborative robotics, compliant joints or force sensors might be integrated so the arm can stop if it contacts a person. While most gesture-controlled arm projects in research do not implement full industrial safety, they often use relatively lightweight arms or virtual simulations to mitigate risk during development.

In summary, integrating a robotic arm for gesture control involves balancing mechanical complexity with the intended functionality. A high-DOF, sensor-rich arm can closely emulate a user's arm pose (improving intuitiveness), but simpler hardware might suffice for tasks like pointing or grabbing where a subset of motions are needed. Researchers thus tailor hardware choices to their goals: for example, a gesture-controlled fruit-harvesting robot might use a 6-DOF arm mounted on a mobile base for orchard maneuverability, whereas a desktop telemanipulation system might use a small 4-DOF arm for simplicity. As gesture interfaces advance, it's expected that modular, 3D-printable designs will allow rapid prototyping of arms, and improvements in actuators (like variable stiffness motors) will further enhance the synergy between human gestures and robot movement.

4.2 Embedded Systems & Communication

Between the human gesturing and the robotic arm moving, there is an entire control pipeline facilitated by embedded systems and communication protocols. At the user side, any wearable sensors (glove, EMG, etc.) typically interface with a microcontroller. For instance, a custom ESP32-based board was used in one glove to read flex sensor resistances and transmit the data wirelessly. Microcontrollers like Arduino or ESP32 are popular for their ease of use in reading analog/digital sensor inputs and performing simple computations (e.g. smoothing sensor values). They can then send the gesture data to the main controller via Bluetooth, Wi-Fi, or wired serial (USB/UART). In the case of camera-based systems, the "embedded" part might simply be a PC or single-board computer that the camera feeds into – for example, a Raspberry Pi or an NVIDIA Jetson might run the vision processing and also control the robot.

On the robot side, the arm's motors are often controlled by either the same microcontroller or a dedicated controller. Small hobby servos can connect directly to an Arduino, whereas more complex arms have their own drive electronics and may accept high-level commands. The communication between the gesture recognition module and the robot controller must be real-time or near-real-time. Some systems use a wired USB or serial link for reliability (especially in lab settings). Others go fully wireless: one project demonstrated a glove sending commands via a LoRa wireless module to a distant robotic arm, showing the feasibility of remote gesture control over long distances (albeit with some latency).

In more advanced setups, middleware like ROS (Robot Operating System) is employed to handle communication and control. ROS provides a flexible framework where the gesture sensor publishes a stream of target joint angles or end-effector positions, and the robotic arm's control node subscribes to these and executes motion planning. For example, in one study the user's hand pose (from an RGB-D camera) was sent into a ROS environment where MoveIt! handled inverse kinematics and motion execution for the simulated Panda arm. Using ROS can simplify complex tasks (trajectory smoothing, obstacle avoidance) and make the system more modular. However, it requires a computer in the loop; thus, many systems use a PC + microcontroller hybrid: the PC (or SBC) runs the heavy algorithms (vision, AI, high-level planning) and sends simple commands to a microcontroller that directly drives the arm's motors.

Communication protocols vary: common ones include Bluetooth Low Energy for short-range wearables (e.g. a smartwatch used as a gesture controller for an industrial robot transmitted orientation data to a PC via Bluetooth), and CAN bus or Ethernet for interfacing with robot joints (many commercial arms have CAN bus networks for their motors). In research prototypes, simpler protocols like I2C, SPI or UART might connect sensor modules, while high-level commands go over USB serial or wireless. Ensuring low latency and synchronization is key – a delay of even 100–200 ms between a gesture and robot response can be noticeable to users and degrade the feeling of control. One reason many studies keep processing on a local machine (as opposed to cloud processing) is to minimize such latency.

System architecture typically follows a pipeline: sensor input → data processing → command output. As an example, the agricultural gesture control system had four main software programs working in concert: (1) firmware on the ESP32 reading glove sensors, (2) a data processing server program that receives sensor data (and OptiTrack motion data) and pre-processes it, (3) a deep learning program that classifies the gesture from the sensor stream, and (4) the robotic arm control program that receives the recognized gesture or target position and sends movement commands to the arm. These components were networked via Wi-Fi and local sockets, highlighting the distributed nature of the system. Many robotics setups similarly distribute tasks between threads or devices to meet real-time demands.

In summary, a robust embedded system and communication design is crucial to tie the gesture interface to the robotic arm. Simpler projects might be achieved with a single microcontroller (reading a few sensors and controlling a few servos directly). But as complexity grows – multiple sensors, vision processing, multi-DOF arms – designers partition the system into sensing, computing, and actuation subsystems linked by reliable communication channels. With the advent of powerful microcontrollers and SBCs, even computationally heavy tasks like running a CNN can now be done on or near the device, making fully untethered gesture-controlled robots increasingly feasible.

4.3 Design Considerations (Cost, Accuracy, Complexity)

Designing a gesture-controlled robotic arm involves trade-offs between cost, accuracy, and system complexity. One major factor is the selection of sensors and hardware within budget constraints. High-end motion capture systems or instrumented gloves (like the HaptX VR glove or Manus Prime glove) can provide very accurate tracking, but at a steep cost – for instance, a HaptX Gloves kit can cost around \$5,500. In contrast, a custom-built glove with flex sensors can be made for roughly \$100 in parts, but might sacrifice some precision and durability. Table 1 illustrates a comparison: the Manus Prime glove (which uses IMUs and bending sensors with a vision-based calibration) costs \$4,000, whereas a simple IMU-only glove can be \$200. Researchers must decide if the incremental accuracy of expensive gear is warranted by the use-case. For a lab prototype aimed at concept demonstration, low-cost components are often chosen to maximize accessibility. But for a deployable assistive device (e.g. a prosthetic interface), the robustness and reliability of higher-grade sensors might justify the cost.

Another aspect is algorithmic complexity vs. performance. Using sophisticated AI models can improve accuracy and allow more complex gestures, but requires more processing power and careful tuning. Simpler algorithms (like thresholding a flex sensor to detect a fist) are easy to implement on microcontrollers but limit the nuance of gestures recognized. One theme in recent studies is the attempt to keep systems as simple as possible while meeting performance needs – an engineering principle often phrased as avoiding "over-engineering." For example, if the goal is to control a robot gripper (open/close) and one arm joint up or down, a single IMU on the hand might suffice, whereas tracking every finger joint and arm pose (a much harder problem) would be unnecessary. However, if a project aims for general-purpose teleoperation of a robot, then a more complex full-hand tracking with many sensors or advanced vision is justified.

Calibration and usability also factor in. A highly complex system that requires lengthy calibration (e.g. attaching many markers to a user or calibrating each sensor each session) may discourage practical use. One gap identified in literature is the lack of focus on quick calibration or auto-calibration – many prototypes assume expert setup. Designs that minimize calibration (using self-calibrating algorithms or factory-calibrated sensors) enhance usability. For instance, the use of an Intel RealSense depth camera that is factory-calibrated helped one system avoid manual camera calibration, speeding up deployment.

In terms of mechanical design, more complex arms (with more DOF or advanced end-effectors) naturally cost more and are harder to control. One study explicitly notes that a high-DOF arm "offers significant flexibility" but that the selected depth sensor's accuracy became the limiting factor in overall precision. This highlights how the chain is only as strong as its weakest link: pouring resources into a 7-DOF arm is moot if your gesture sensing cannot reliably command all those DOFs with matching precision. On the other hand, an under-actuated or low-DOF arm might not fully translate a user's intent (leading to a less intuitive experience). Thus, designers often iterate to find a sweet spot where the fidelity of gesture input matches the capability of the robot output.

Finally, scalability and portability are considerations. Using common platforms like Arduino, Raspberry Pi, and standard sensors ensures that others can replicate and extend the system easily. Open-source designs and 3D-printed parts also make the research more accessible. Many recent papers share their hardware designs or code, promoting a community-driven improvement of these systems. This is essential for the field to progress toward real-world adoption – not only should a system work in a paper, but it should be reproducible and maintainable in practice.

In conclusion, the design of gesture-controlled robotic arms is a multi-objective optimization between cost, accuracy, and complexity. The trends suggest a push towards lower-cost solutions that still meet accuracy requirements, enabled by clever use of AI to compensate for cheaper hardware. There is also a call for designing with the end-user in mind: a system with fewer parts and straightforward setup is more likely to succeed outside the lab. The next section will delve into how these design choices play out when real users test the systems.

5. User Testing and Human-Centered Evaluation

A crucial part of developing gesture-controlled systems is evaluating them with human users – assessing not only technical performance but also usability, comfort, and intuitiveness. We review how studies have conducted such evaluations, who the participants have been, what findings emerged (both qualitative and quantitative), and what gaps remain in making these systems truly human-centered.

5.1 Evaluation Methods

Researchers have employed a variety of methods to evaluate gesture-controlled robotic arms, often combining objective task performance measures with subjective user feedback. A common approach is to define specific tasks for participants to perform using the gesture interface and then measure metrics like success rate, completion time, and error rate. For example, a task might be to use hand gestures to pick up a set of objects with the robot arm and place them in a target area. Objective measures could include how long it takes to complete the task compared to a baseline (like using a traditional control), how accurately

the robot end-effector reaches target positions (perhaps measured in centimeters of error), or how many mistakes occur (such as false trigger gestures or drops).
In addition to task performance, usability and user experience are gauged through questionnaires and observations. Many studies utilize standard surveys like the System Usability Scale (SUS) or custom questionnaires asking about ease of use, comfort, fatigue, and user satisfaction. For instance, after trying a Leap Motion-based control, users might be asked to rate how natural the control felt and whether they experienced arm fatigue from holding their hand in the sensor's view. Some evaluations include a NASA TLX (Task Load Index) to assess perceived workload (mental, physical, temporal demand, etc.), especially if comparing multiple control methods. Interviews or think-aloud protocols are sometimes used to gather qualitative insights, such as which gestures users found confusing or how they adapted their strategy to complete the task.
Another aspect of evaluation is the learning curve. Researchers might measure how performance improves over repeated trials as users become familiar with the system. A steep learning curve (performance rapidly improving then plateauing) indicates the system can be learned quickly. For example, in an assistive scenario with older adults controlling a bathing robot via gestures, initial sessions were slow, but with a brief training period, the older users became significantly more adept at issuing the correct gestures, indicating training can improve outcomes . Some studies explicitly include a training session and then measure performance before and after training to quantify learning effects.
Comparative evaluations are also common: a gesture interface might be compared against another interface (e.g., compare a glove-based control vs. a vision-based control for the same task) or against manual operation as a benchmark. In one case, researchers compared a Leap Motion controller to a traditional gamepad for controlling a robot arm, finding that while the Leap interface was more intuitive for novices, expert users could achieve tasks faster with the gamepad – highlighting trade-offs in efficiency vs. intuitiveness . Such comparisons help identify the contexts in which gesture control truly adds value.
Lastly, many technical papers perform systematic accuracy evaluations without external users: e.g., move a hand to known positions or perform predefined gestures and log how accurately the system detects them. While not a full "user study," these controlled tests (using either the experimenters themselves or robotic test rigs) provide baseline performance data (like confusion matrices for gesture classification) that can be correlated with user outcomes.
In summary, evaluation methods in this field are multi-faceted, combining task performance metrics (speed, accuracy) with usability metrics (ease, comfort, workload). The goal is to ensure not only that the robot moves correctly in response to gestures, but that human users can effectively and happily use the system. The next subsections discuss who those users typically are and what findings have emerged from these evaluations.

5.2 Participants

Across the reviewed studies, participants in user tests have often been relatively small groups of tech-savvy individuals, typically young adults (such as university students or lab personnel). For example, many prototype evaluations involve on the order of 5–15 participants in their twenties, which is convenient but not fully representative of all potential users. These participants are usually able-bodied and have no prior experience with the specific system (to test ease of learning). Demographically, there is often a lack of diversity – few studies report involving older adults, children, or people with disabilities in their experiments. One notable exception was a study focusing on older adults using a gesture-controlled assistive bathing robot, wherein about 18 seniors (average age ~75) were recruited. Their inclusion provided valuable insights into age-related challenges (like tremor or slower reaction times) that younger users might not exhibit.

For prosthetic arm control research, it's still common that initial tests use able-bodied subjects performing gestures to control a prosthetic simulator (perhaps by EMG on intact limbs) – meaning actual amputees are not always involved in early-stage testing. This is a gap, as the end-users (amputees) may have different needs or signal characteristics. A few advanced studies do include amputee participants in late-stage trials, but these are relatively rare and often limited to case studies (1–3 amputee users) due to recruitment difficulty.

Participant limitations are frequently acknowledged. Most studies have small sample sizes, which limit the statistical power of the findings. Also, convenience sampling (recruiting university students, etc.) means the users are not always the target population. For an industrial robot interface, ideally one would test with actual factory workers; for an assistive arm, test with people who have motor impairments. In practice, many papers have not yet taken that step, primarily evaluating with proxy users in a lab. This leaves open questions about how the systems perform for those real target users. For example, gestures that are easy for a 25-year-old might be difficult for an elderly person with arthritis.

Researchers are aware of these limitations and often note them as future work: expanding participant demographics, including novices and experts, or including those with relevant impairments. A positive trend is that some recent projects have begun co-designing with end-users – e.g. involving amputees in choosing which gestures should control which prosthetic hand function – to ensure the system matches user expectations .

In summary, participant groups in existing studies have been skewed toward young, able-bodied users, and sample sizes are small. This is understandable in early prototype evaluations but highlights a need for more inclusive and extensive user testing as the technology matures. Broader testing will be essential to verify that gesture-controlled robotic arms can accommodate different ages, body abilities, and skill levels.

5.3 Findings from Usability Testing

Despite the varied setups and participant groups, several common findings emerge regarding the usability of gesture-controlled robotic arms:

- **Intuitiveness and Ease of Use:** Most studies report that users generally find gesture-based control natural and intuitive once they understand the gesture set. The direct mapping between their physical movement and the robot's movement tends to shorten the learning curve. For instance, users controlling a robot arm via hand motions (as tracked by a depth camera) could often start performing simple pick-and-place tasks with minimal instruction, something much harder to do with a complex joystick interface. This intuitiveness is a major advantage – one paper noted that hand gestures provide a "straightforward means for control without the need for significant training" of operators. In comparative feedback, novices often prefer the gesture interface to a traditional control for basic tasks, because it "just makes sense" to move your hand and see the robot follow.
- **Fatigue and Ergonomics:** A consistent concern is user fatigue, especially for vision-based systems that require the user to hold their arms in mid-air. Some studies found that after 10–15 minutes of continuous use, users felt tired in the shoulder or arm. The Leap Motion interface can cause fatigue if the user must keep their hand above the sensor for long durations, a phenomenon informally dubbed the "gorilla arm" effect. However, clever interface design can mitigate this: for example, placing the sensor so the user can rest their elbow on a surface, or using a gesture to allow "clutching" (pausing control to relax the arm). In the agricultural arm study, the developers emphasized an ergonomic design

of the glove and an interface that mimics natural hand positions to minimize strain. Some user tests also reported that glove-based control can cause hand fatigue or sweating after extended use, due to the physical glove. Lighter, fingerless glove designs (just with tape-on sensors) are being tried to reduce this issue.

- **Accuracy and Reliability Perceived:** Users generally appreciate when the robot accurately follows their intended gesture. High accuracy of recognition (e.g. >95%) was noted in technical terms, but what matters to users is that mistakes or misinterpretations are rare. Findings show that even a 90% recognition rate can be frustrating if those 1 in 10 errors happen at critical moments. Participants often commented on any occurrence of the robot doing something they didn't mean. For example, if a vision system occasionally misreads a waving gesture as a "grasp" command, users lose trust and might slow down or exaggerate gestures to compensate. Thus, subjective perception of reliability is key. Systems that fuse sensors to improve reliability (as discussed earlier) often get positive user feedback like "the control felt responsive and error-free." In the hybrid glove+OptiTrack system, participants were able to guide the robot to mimic complex hand movements with very few missteps, which was reflected in high user satisfaction scores.
- **Gesture Set and Mental Load:** The choice of which gestures to use and how many distinct gestures the system recognizes can impact usability. Simpler gesture sets (e.g., using just a few intuitive gestures like open hand = open gripper, fist = close gripper) are easier for users to remember and resulted in fewer errors in multiple studies. When systems try to recognize a large number of gestures, users may confuse them or the system may have more false positives. One human factors study noted "inconsistencies between developers and users in preferred hand gesture poses" – meaning the gestures designers assumed would be easy were not always the ones users found natural. This finding suggests that involving users in defining the gesture vocabulary (user-centered design) leads to better outcomes. Training users on the gesture commands was shown to significantly improve performance in at least one case (older adults controlling a bath robot), but ideally the system should be as guessable as possible. Lower mental load and memory burden correlate with better task performance and user satisfaction.
- **Latency and Feedback:** Users are sensitive to control latency. If the robot lags notably behind their gesture, it can cause overshooting or a feeling of disconnect. Most research systems aim for latency under about 100 ms; when achieved, users reported the control as "real-time" and satisfying. Providing some form of feedback to the user also helps – for example, visual feedback on what gesture the system detected (maybe via a HUD or a light) or haptic feedback in a glove. Not many studies implemented feedback beyond the robot's motion itself, but those that did found it improved user confidence, especially in complex gesture sets (confirming the system recognized a "stop" gesture, for instance, by a beep).

In general, user testing findings validate that gesture control can be highly effective for certain tasks, but they also highlight areas to improve: reducing user fatigue, ensuring consistent accuracy, and aligning the system's gesture-language with human preferences. Users tend to be enthusiastic about the concept – often describing the experience as "futuristic" or "like controlling with my mind" – but they also point out pragmatic issues which researchers are actively addressing (like accidental activations, need for personalization, etc.). These findings underscore the importance of iterative design: testing with users, learning from their experiences, and refining the system.

5.4 Gaps Identified

While significant progress has been made, the literature points to several gaps and underexplored areas in development and testing:

- **Underrepresented User Groups:** As noted in section 6.2, many systems have not been tested with the actual end-user populations that would benefit most. There is a gap in understanding how gesture-controlled arms perform for older adults, children, or users with disabilities. For example, an interface might need adaptation for those with limited range of motion or tremors. Few studies include such participants, so designs may be biased towards the capabilities of young, able users. Moving forward, researchers call for more inclusive testing – for instance, involving amputees early in the design of prosthetic arm controllers, or evaluating industrial gesture controls with workers of varying skill levels. Ensuring inclusive design will likely improve these systems' robustness and usefulness for everyone.
- **Real-World Trial and Long-Term Use:** Most evaluations have been in controlled lab settings or very short-term trials. There is a notable lack of longitudinal studies examining how users adapt to gesture control over days or weeks, and how reliable the system is over extended use. Real-world environments introduce noise: lighting changes, cluttered backgrounds, or the user multitasking. One paper explicitly mentioned the need to integrate the system into a real industrial environment to gather insights on its performance outside the lab. The absence of field trials means we don't fully know how these interfaces hold up under continuous daily use – for example, will a factory worker still find a gesture interface comfortable after 8 hours? Will an assistive robot controlled by gestures be practical in a home with various distractions? These remain open questions.
- **Robustness and Error Handling:** Many prototypes assume cooperative conditions and do not robustly handle gesture recognition failures or robot errors. If a user's gesture isn't recognized, or the robot fails to execute a command (perhaps due to a physical constraint), how is this conveyed and recovered? Ideally, a system should have fail-safes – e.g., if input is lost, the robot should safely stop. However, few papers detail such safety measures. The focus has been on achieving functionality, and only recently is attention shifting to reliability and safety in realistic conditions. The gap here is transitioning from "it works most of the time" to "it fails gracefully and rarely." Some hybrid approaches attempt to address this by using multiple sensors (if one fails, another takes over), but comprehensive solutions (and standards for evaluation of failure modes) are not yet established.
- **Standardized Metrics and Benchmarks:** The field currently lacks widely accepted benchmarks for comparing gesture-controlled systems. Each study often defines its own tasks and metrics, making it difficult to quantitatively compare systems. There is a gap in terms of a standard evaluation framework – for example, a set of benchmark tasks or a common set of gestures that all systems could be tested on for performance and usability. Without this, it's hard to say definitively which technique is "best" or how much progress has been made in numeric terms. Establishing shared benchmarks (perhaps akin to how computer vision has ImageNet, etc.) would help drive the field forward.
- **Human Factors and Co-Design:** Another gap is the limited involvement of end-users in the design phase (not just testing phase). The inconsistency noted between developer-chosen gestures and user-preferred gestures suggests a need for more co-design: having users help define gesture vocabularies, layouts of sensors, and features of the system from the ground up. While some human–computer interaction (HCI) researchers have started to explore this, many engineering-driven projects still pick gestures ad hoc. Bridging this gap could improve intuitiveness and adoption.

- **Multimodal Interaction:** Currently, most systems focus on gestures alone. There is an opportunity (and thus a gap currently) to combine gestures with other modalities like speech or gaze for a richer interaction. For example, a user might gaze at an object and perform a grab gesture to command the robot to fetch it – combining eye-tracking with hand gesture. Such multimodal systems could overcome some limitations of using gestures in isolation (like reducing the number of gestures needed). Few studies in our 2015–2025 range have fully explored this integration.

In summary, the gaps identified revolve around expanding the scope of testing (more users, real settings, longer term) and improving the systems' resilience and user-centeredness. Addressing these gaps is essential before gesture-controlled robotic arms can move from prototype stage to widespread real-world deployment. The comparative analysis in the next section, along with the future directions, will further illuminate how current research is beginning to fill these gaps and what remains to be done.

6. Comparative Analysis of Studies

To illustrate the diversity of approaches and highlight pros/cons, we compare a selection of representative recent studies on gesture-controlled robotic arms (2019–2025). Table 1 provides an overview of four example systems, including their input methods, the type of robotic arm used, testing scenario, and key advantages and limitations noted.

Table 1. Comparison of selected gesture-controlled robotic arm studies (2019–2025).

Study (Year)	Gesture Input Method	Robotic Arm & Scenario	Pros (Advantages)	Cons (Limitations)
Hameed et al. (2021) – Lab demo using Leap Motion	Vision-based (Leap Motion IR hand tracker)	5-DOF robotic arm (simulated surgical tool) in lab tests	High precision control noted; very natural hand mapping (position & orientation)	Limited range (hand must stay within sensor view); not tested outside of indoor lab environment
Tikam et al. (2020) – Low-cost glove prototype	Sensor-based (wearable glove with flex sensors + accelerometer)	6-axis robotic arm (Arduino-controlled), basic pick-and-place tasks	Inexpensive setup (DIY components); straightforward to learn (few simple gestures)	Lower accuracy and some drift observed (due to sensor noise); glove calibration needed periodically
Yu et al. (2024) – Agri-Arm system	Hybrid: bending sensor glove and OptiTrack motion capture (markers)	6-DOF mobile robotic arm, tested in field-like harvesting scenario	Very high precision (≈1 cm error) and 96% gesture recognition accuracy; performed well in real-time remote operation	Requires expensive motion-capture setup (OptiTrack cameras); vulnerable to marker occlusion outdoors; hardware cost limits practicality

Study (Year)	Gesture Input Method	Robotic Arm & Scenario	Pros (Advantages)	Cons (Limitations)
			(mimics human arm)	
Angelidis & Bampis (2025) – Assembly-line controller	Vision + AI: RGB-D camera with AI hand-pose tracking (no wearables)	7-DOF Franka Emika Panda (simulation for small factory tasks)	Intuitive, user-friendly control (no training needed); no user gear required; AI allows robust tracking of hand joints	Complex system setup (ROS, AI model, calibration steps); tested only in simulation (no physical robot in factory yet); depth sensor errors can affect precision

Discussion

These studies showcase how different input modalities and design choices fare in practice. Hameed et al.'s Leap Motion approach achieved fine control with a modest 5-DOF arm and was praised for intuitiveness, but it illustrates the common limitation of vision-based systems: the user must stay within a defined workspace (here, within the Leap Motion's ~0.5 m range) and environmental robustness wasn't proven. Tikam et al.'s glove, on the other hand, represents a minimalistic hardware approach – highly accessible and easy to replicate – yet it suffered from accuracy issues like drift and required recalibration, which could impede usability in longer tasks. Yu et al.'s hybrid agricultural arm demonstrates the power of combining methods: it excelled in performance (both fast and accurate, even in a semi-real setting), highlighting that multi-sensor fusion can overcome individual weaknesses. Its downside is the complexity/cost of the setup (OptiTrack is state-of-the-art but not deployable on a farm without significant investment). Finally, Angelidis & Bampis (2025) show a cutting-edge vision+AI system aligned with Industry 5.0 concepts – very user-centric (no wearables, AI doing the heavy lifting) and flexible with a high-DOF robot. The trade-off there is system complexity and the fact that it's still mostly validated in simulation; real-world deployment will need to confirm its reliability and address any physical depth sensor limitations.

Across these comparisons, a trend emerges: hybrid and AI-enhanced systems tend to achieve the best performance (accuracy, natural feel) but at the cost of complexity and sometimes expense, whereas simpler wearable-based systems are affordable and easier to implement but may need further refinement to match the performance and comfort levels users desire. Each approach has its niche – for instance, a low-cost glove might be perfect for a classroom or hobby project, while a vision+AI system might be favored in a smart factory where workers can interact with multiple robots freely. The pros/cons listed provide insights that feed into future research directions, as discussed next.

Future Directions

Building on current trends and gaps, several promising future directions have been identified for gesture-controlled robotic arms:

- **AI-Powered Predictive Control:** The next generation of systems will likely leverage AI not only for recognizing gestures but also for predicting user intentions and smoothing control. This could involve machine learning models that adapt to an individual user's motion patterns over time (personalization) or that predict the user's next gesture to reduce apparent lag. For example, if a user is repetitively picking and placing, the AI could anticipate the likely next move and pre-position the robot arm, making control feel more responsive. Enhanced predictive algorithms could also filter out erratic hand tremors or involuntary movements, which is crucial for users with motor impairments. Researchers are exploring techniques like deep reinforcement learning to allow the robot to interpret higher-level intent rather than just mirroring motions. Ultimately, AI could enable more context-aware gesture control – the robot might understand what the user is trying to achieve and assist actively (akin to an intelligent assistant), rather than strictly following literal hand motions.

- **Low-Cost, Prosthetic-Grade Arms:** A significant future direction is translating these gesture control technologies into affordable prosthetic and assistive devices. This involves not just refining the gesture recognition (which for prosthetics often means improving EMG decoding or residual limb tracking) but also developing robotic arms that are lightweight, safe, and inexpensive. Advances in 3D printing and open-source hardware are expected to produce prosthetic arms in the sub-$1000 range that can be controlled via intuitive gestures or muscle signals. One comprehensive 2023 review noted a surge in using non-invasive sensors (like EMG, ultrasound, inertial) along with intelligent algorithms to improve prosthetic hand control. The goal is to "revolutionize prosthetic hand control" by making it more natural and reliable. This includes improving the portability of gesture sensors (perhaps developing all-in-one armbands or patches that provide multimodal sensing) and ensuring the control system can run on battery-powered embedded hardware. Success in this direction would directly benefit amputees and those who rely on assistive robot arms for daily activities, bringing the cost and complexity down to practical levels without sacrificing performance.
- **AR/VR Integration for Training and Feedback:** Incorporating augmented reality (AR) and virtual reality (VR) opens new possibilities for both user training and real-time feedback. AR could be used to visually guide users in performing gestures or to highlight to the user what the robot "sees" and understands – for example, projecting a holographic hand model that mirrors the detected gesture, so the user knows if their gesture was recognized correctly. This kind of feedback can reduce the uncertainty in gesture control. Training in VR is another exciting avenue: a user could practice controlling a virtual robotic arm in a safe simulated environment (perhaps a game-like scenario) to gain proficiency before using a real robot. VR can also be used for telepresence, where the user sees through the robot's eyes and controls it with natural gestures, feeling as if they are embodying the robot – this could be useful for remote surgery or hazardous environment manipulation. Some initial studies in HRI have used VR to acclimate users to robots and have found it improves subsequent real-world interaction. We anticipate AR/VR tools will become standard in developing gesture interfaces, both to accelerate user onboarding and to provide ongoing cues (visual or even haptic cues) that enhance control accuracy and user confidence.
- **Inclusive and Adaptive Design:** The future of gesture-controlled arms will place heavy emphasis on inclusive design, ensuring systems can accommodate a wide range of human diversity. This includes making gesture recognition adaptive to different skill levels – for instance, an "adaptive gesture library" that might simplify the required gestures if it detects a user has limited dexterity. It also means supporting left-handed and right-handed users equally, and allowing customization of gestures (so a user can choose a gesture that is comfortable for them for a given command). Culturally, gestures can have different meanings or comfort levels, so localization might be considered for global products.

Another facet is designing for accessibility: e.g., using larger, slower gestures for users with cognitive impairments who need more time, or incorporating confirmations to avoid false triggers. The user-centered iterative design process will be crucial – involving target users in testing early prototypes, as several authors have advocated . In the future, we expect to see more studies like Werner et al. (2020) who specifically looked at older adults and adapted the interface to their needs. The end goal is gesture-controlled robotic arms that are truly plug-and-play for anyone, with the system smart enough to adjust to the user, rather than the user having to adapt to the system.

- **Real-World Deployment and Longitudinal Studies:** As systems mature, a key future step is deploying them in real operational environments for extended periods. For industrial use, this might mean month-long trials on a factory floor, measuring productivity, safety incidents, and worker feedback when using gesture-controlled cobots. For assistive use, this could involve users taking a gesture-controlled helper robot home and logging usage patterns and outcomes over weeks. These longitudinal studies will provide insights into durability (do sensors degrade or do recognition rates drop over time?), and into behavioral adaptation (do users develop shortcuts or experience fatigue with prolonged use?). They will also surface unanticipated issues – for example, how do these systems perform with background movement (other people) or in outdoor conditions? Early hints from simulation-based work suggest adding additional sensors or fallback modes could help (as noted by Angelidis & Bampis, combining multiple cameras or ML models might retain low cost while improving robustness). Future deployments will test such ideas. The ultimate vision is that gesture-controlled robotic arms move from lab demos to everyday tools – in smart factories, rehabilitation centers, and even households. Achieving that will require this real-world validation phase, which we foresee happening increasingly in the next few years.

In summary, the future of gesture-controlled robotic arms lies in making them smarter, more user-adaptive, and more integrated into real tasks. AI will play a big role in predictive and context-aware control, while AR/VR will enhance user interaction and training. Importantly, a user-centered philosophy will drive design, ensuring these systems are inclusive and practical. Each of these directions addresses current limitations: improving accuracy and ease (AI), lowering cost and increasing impact (prosthetics focus), enhancing usability (AR/VR, inclusive design), and proving reliability (real-world trials). If researchers and engineers successfully navigate these directions, gesture-based control could become a mainstream, perhaps even ubiquitous, modality for human-robot interaction.

Conclusion

Gesture-controlled robotic arms have rapidly evolved from experimental prototypes to sophisticated systems that leverage cutting-edge technology for natural HRI. This secondary study reviewed developments in both the enabling technologies and the user testing of such systems. We found that sensor-based methods (like gloves and EMG) and vision-based methods (like depth cameras and Leap Motion) each offer unique advantages, and the state-of-the-art often combines them with AI algorithms to achieve high accuracy and an intuitive feel. On the robotic arm side, advances in hardware (e.g. affordable multi-DOF arms, better actuators) and software (ROS-based control, improved kinematics) have provided solid platforms for gesture interfaces.

User evaluations across studies consistently underscore the importance of user-centered design: systems that align with users' natural gestures and provide sufficient feedback tend to score well in usability, whereas those requiring cumbersome setup or causing fatigue face adoption barriers. A key takeaway is that technical performance alone is not enough – a gesture interface must be evaluated in terms of how real people interact with it. Encouragingly, many studies report that with minimal training, users can effectively control robots via gestures and often prefer it for its intuitiveness and directness. At the same time, our review highlighted gaps such as the need for more diverse participant testing and more real-world trials to ensure these systems robustly meet user needs outside lab conditions.

The findings from this secondary study lead to several recommendations. Developers of gesture-controlled arms should strive to involve end-users early and throughout the design process – for instance, to choose an appropriate gesture set and to identify potential accessibility issues. It's also recommended to incorporate multiple sensing modalities and error-handling mechanisms to make systems fault-tolerant and safe for real deployment. Researchers should work towards standardizing evaluation protocols (task benchmarks, common metrics) to collectively drive progress. Finally, interdisciplinary collaboration will be vital: combining insights from robotics, computer vision, human factors, and even sociology (for understanding user acceptance) will yield the most robust designs.

In conclusion, gesture-controlled robotic arm technology holds great promise for making human-robot interaction more natural, efficient, and inclusive. From enabling a factory worker to intuitively guide a robot coworker, to allowing an amputee to control a prosthetic arm with the ease of thought, the applications are far-reaching. The past decade has seen substantial progress in the core technologies and initial usability demonstrations. The next steps involve refining these systems with a human-centered lens – ensuring they work reliably for all users, in real environments. By emphasizing user testing, addressing current limitations, and embracing emerging innovations like AI personalization and AR feedback, researchers can develop gesture-controlled robotic arms from a novel interface into a practical tool that enriches human capabilities and safety. The ongoing convergence of low-cost sensors, powerful algorithms, and user-driven design makes us optimistic that in the near future, controlling robots through natural gestures will move from the realm of demos and lab studies to an everyday reality in workplaces, clinics, and homes. The importance of keeping users at the center – making systems that adapt to people, rather than expecting people to adapt to systems – cannot be overstated as we push toward that reality.

References (2019–2025)

1. Aldabbagh, A., & Alrasheed, A. (2021). *Gesture-controlled robotic arm utilizing OpenCV.* In *Proceedings of the 2021 IEEE HORA (Human-Computer Interaction, Optimization and Robotic Applications)* (pp. 1–6). IEEE. https://doi.org/10.1109/HORA52670.2021.9461389
2. Angelidis, G., & Bampis, L. (2025). *Gesture-controlled robotic arm for small assembly lines. Machines, 13*(3), 182. https://doi.org/10.3390/machines13030182
3. Gautam, A., Ferreira, A., & Dunai, L. (2025). *Revolutionizing prosthetic hand control using non-invasive sensors and intelligent algorithms: A comprehensive review. Computers & Electrical Engineering, 102*, 108200. https://doi.org/10.1016/j.compeleceng.2025.110094
4. Li, W., Shi, P., & Yu, H. (2021). *Gesture recognition using surface electromyography and deep learning for prostheses hand: State-of-the-art, challenges, and future. Frontiers in Neuroscience, 15*, 621885. https://doi.org/10.3389/fnins.2021.621885
5. Li, Y., Lou, J., Cai, Z., Zheng, P., Wu, H., & Wang, X. (2024). *An interactive gesture control system for collaborative manipulator based on Leap Motion Controller. Advances in Mechanical Engineering, 16*(5), 16878132241253101. https://doi.org/10.1177/16878132241253101

6. Lin, C.-S., Chen, P.-C., Pan, Y.-C., Chang, C.-M., & Huang, K.-L. (2020). *The manipulation of real-time Kinect-based robotic arm using double-hand gestures. Journal of Sensors, 2020*, Article 8819236. https://doi.org/10.1155/2020/8819236
7. Uba, J., & Jurewicz, K. A. (2024). *A review on development approaches for 3D gestural embodied human–computer interaction systems. Applied Ergonomics, 121*, 104359. https://doi.org/10.1016/j.apergo.2024.104359
8. Wang, X., Shen, H., Yu, H., Guo, J., & Wei, X. (2023). *Hand and arm gesture-based human–robot interaction: A review.* In *Proceedings of the 6th International Conference on Algorithms, Computing and Systems (ICACS 2022)* (pp. 1–7). ACM. https://doi.org/10.1145/3564982.3564996
9. Werner, C., Kardaris, N., Koutras, P., Zlatintsi, A., Maragos, P., Bauer, J. M., & Hauer, K. (2020). *Improving gesture-based interaction between an assistive bathing robot and older adults via user training on the gestural commands. Archives of Gerontology and Geriatrics, 87*, 103996. https://doi.org/10.1016/j.archger.2019.103996
10. Yu, Z., Lu, C., Zhang, Y., & Jing, L. (2024). *Gesture-controlled robotic arm for agricultural harvesting using a data glove with bending sensor and OptiTrack systems. Micromachines, 15*(7), 918. https://doi.org/10.3390/mi15070918