

Assisted Robot Arm with Voice Recognition for Digital Healthcare

Ting Zhang^{1*}, Oanh Phan², Jiang Lu³,

Ting Zhang-Program of Computer Science, University of Houston Downtown, Houston, USA¹

Oanh Phan-Program of Computer Science, University of Houston Downtown, Houston, USA²

Jiang Lu-Program of Computer Engineering, University of Houston Downtown, Houston, USA³

Abstract—Healthcare systems face increasing pressure from nursing shortages, growing patient-care demands, and the substantial amount of time healthcare professionals spend performing routine and repetitive bedside activities. These challenges can increase caregiver workload and reduce the time available for clinical assessment, decision-making, and direct patient interaction. Patients with limited mobility may also depend on caregivers for relatively simple activities, such as retrieving objects, obtaining drinking water, or interacting with their immediate environment, which can lead to delayed assistance when healthcare personnel are occupied with higher-priority responsibilities. Motivated by these challenges, this study investigates the use of collaborative robotics and intelligent natural-language interaction to provide supplementary assistance for predictable and repetitive patient-care tasks. An intelligent robotic patient-assistance framework is developed by integrating collaborative robotic manipulation, voice-based natural-language interaction, and non-contact physiological monitoring. The system employs a UFACTORY xArm7 collaborative manipulator to perform representative bedside assistance tasks, including cleaning a designated area, pouring water, delivering medicine, and measuring body temperature. A Python-based software architecture coordinates robot control, task execution and user interaction. To provide hands-free and intuitive control, Faster-Whisper is used for continuous speech-to-text processing, while a Large Language Model (LLM) interprets the transcribed commands and maps different natural-language expressions to predefined robotic tasks.

Keywords—Healthcare; Robot Arm; LLMs; automation; voice recognition; healthcare innovation

I. INTRODUCTION

Healthcare is an ever-evolving field that continuously pursues innovative approaches to improve patient outcomes and enhance the efficiency of healthcare operations. Rapid technological advancements have created new opportunities to transform healthcare delivery, with robotics emerging as a promising solution for supporting clinical tasks, improving workflow efficiency, and advancing the overall quality of patient care.

A. Challenges of Traditional Healthcare and Patient Care

Traditional healthcare systems depend heavily on human healthcare professionals to perform both clinical and routine patient-care activities, creating significant challenges as patient populations grow and workforce shortages persist. The global shortage and uneven distribution of nursing personnel can

increase individual workloads and limit the time available for direct patient interaction and continuous monitoring [1]. High workloads and repetitive responsibilities can contribute to physical and psychological fatigue among healthcare workers. Nurse burnout has also been associated with missed care, medication errors, patient safety incidents, and reduced patient satisfaction [2]. Furthermore, inadequate staffing and limited resources may result in delayed or missed nursing interventions [3]. Without robotic manipulators, many repetitive physical tasks must be performed manually, limiting opportunities to automate routine bedside assistance and potentially reducing the time clinicians can devote to tasks requiring professional judgment and interpersonal interaction. At the same time, traditional healthcare information systems generally provide limited natural-language interaction; without large language models (LLMs), which is a new, innovative and powerful technique created from recent years. Recent studies indicate that LLMs have potential applications in patient communication, clinical documentation, and information processing [4], [5]. Therefore, combining large language models with robotic manipulation offers a potential approach for reducing repetitive workload while enabling more intuitive interaction between patients, healthcare personnel, and assistive robotic systems.

1) Nursing shortages and high patient-to-nurse ratios:

Nursing shortages and excessive nursing workloads can contribute to fatigue, emotional exhaustion, reduced job satisfaction, and intentions to leave the profession. A systematic review and meta-analysis found that nurse burnout was associated with poorer patient-safety outcomes, lower perceived care quality, and reduced patient satisfaction [6]. Workforce shortages and burnout may therefore form a reinforcing cycle: insufficient staffing increases workload, elevated workload contributes to burnout, and burnout increases turnover, thereby placing further pressure on the remaining workforce [7], [8].

2) *Economic burden of nursing services:* An adequately staffed nursing workforce represents a substantial but necessary component of healthcare expenditure. The institutional cost of employing a nurse extends beyond base salary and includes health insurance, retirement contributions, paid leave, continuing education, recruitment, orientation, and overtime compensation. Because hospitals and residential-care facilities operate continuously, multiple employees are required to provide uninterrupted coverage for a single position across various shifts [9], [10].

3) *Delayed and missed patient care:* For patients with impaired mobility or upper-limb function, delayed assistance can have consequences beyond temporary inconvenience. A

*Corresponding author.

patient may remain unable to drink, communicate, retrieve an essential object, or adjust the immediate environment until a caregiver becomes available [11], [12]. Some patients may attempt these activities without assistance, potentially increasing the risk of falls, overextension, equipment displacement, or other preventable injuries. Technologies that respond to structured, low-risk requests could therefore improve patient independence while reducing interruptions to nurses performing higher-priority clinical work [14], [13].

4) *Limitations of exclusively human-dependent care:* Direct human interaction remains an essential component of nursing because it enables clinical judgment, emotional support, and individualized patient care. However, relying exclusively on nurses to perform both complex clinical procedures and routine physical tasks presents several operational challenges. Frequent bedside contact increases occupational exposure to infectious diseases despite established infection-control measures [15], [16]. Repetitive activities such as retrieving supplies, repositioning equipment, and assisting with routine patient requests also contribute to physical fatigue and work-related musculoskeletal disorders, which remain common among nursing professionals [18], [17]. Furthermore, the availability of nursing staff is constrained by workload, patient acuity, and emergency situations, making immediate responses to routine requests difficult during busy clinical periods. Consequently, assigning repetitive logistical tasks solely to nurses may reduce overall workflow efficiency and limit the time available for assessment, communication, and direct clinical care. These limitations highlight the potential value of assistive robotic systems that can undertake predictable physical tasks while allowing nurses to focus on responsibilities requiring professional expertise and interpersonal interaction [19].

B. Literature Review of Robotic Assisted Patient Care and Research About LLMs

1) *Existed technologies for robotic assisted patient care:* Healthcare robotics has gradually evolved from autonomous transportation platforms to intelligent assistive systems capable of interacting directly with patients [20]. Early hospital robots were primarily developed for logistics, including the transportation of medication, laboratory samples, and medical supplies, with the objective of reducing caregiver travel and improving operational efficiency. Advances in autonomous navigation, perception, and human-robot interaction have subsequently enabled robots to perform bedside assistance, allowing them to operate safely in clinical environments while collaborating with healthcare professionals [21], [22].

Recent research has increasingly focused on robotic manipulators because physical interaction is required for many patient-care activities. Compared with mobile service robots, robotic arms provide dexterous manipulation that enables direct assistance during activities of daily living [23], [24]. Their applications include object retrieval, assistive feeding, and manipulation of bedside devices for individuals with limited upper-limb function. Shared-autonomy approaches have further improved usability by combining autonomous robot motion with user intention, thereby reducing operator workload while maintaining safe interaction during manipulation tasks [25].

2) *Human-robot interaction and user acceptance:* The successful deployment of assistive robotic systems in healthcare depends not only on technical performance but also on their acceptance by patients and healthcare professionals. Studies have shown that user acceptance is strongly influenced by perceived safety, trust, ease of interaction, and the robot's ability to complement existing clinical workflows rather than replace human caregivers in STEM researches [26]. Although robotic assistants can reduce repetitive physical workload and improve the availability of routine patient assistance, concerns remain regarding privacy, patient dignity, responsibility for system failures, and excessive reliance on automation. These issues highlight the importance of maintaining human supervision during robot operation and ensuring that clinical staff retain full authority over safety-critical decisions [27].

3) *The innovative and powerful technology of LLMs:* Large Language Models (LLMs) are increasingly being explored in healthcare to improve communication and support clinical workflows. They can assist with medical question answering, patient education, and the interpretation of clinical information, allowing healthcare professionals to access relevant knowledge more efficiently [28], [29]. LLMs can also support clinical documentation by generating or summarizing medical text from patient encounters, potentially reducing the administrative workload placed on healthcare providers. In patient-care environments, natural-language interaction may allow patients to express their needs conversationally rather than through predefined commands. When combined with assistive technologies such as robotic systems, this capability can help translate patient requests into structured instructions while maintaining human supervision for safety-critical decisions [30], [31].

4) *LLMs' applications in healthcare about voice recognition:* Large Language Models (LLMs) have emerged as an important artificial intelligence technology capable of understanding natural language and generating context-aware responses. Their ability to interpret complex instructions has supported their growing use in healthcare [28]. LLMs can provide a natural-language interface between users and intelligent healthcare systems, making communication more intuitive. When integrated with robotic systems, an LLM can interpret patient requests, such as "bring me some water," and translate their meaning into structured commands for subsequent robotic execution. This capability may support the development of healthcare assistants that respond more naturally to patient needs, although appropriate validation and human supervision remain necessary for clinical applications [30].

C. Objectives and Scope of the Project

This project introduces the concept of harnessing a collaborative UFACTORY xArm7 as an automated assistant in healthcare. This approach capitalizes on the robot's potential to enhance patient care and efficiency through its role in aiding patient care tasks and conducting vital sign monitoring. We utilized LLMs with the robot arm, to achieve voice recognition for delivery of patient needs. Leveraging the Python programming language for implementation, this project not only underscores the practical fusion of technology with healthcare but also explores the potential paradigm shift in patient care dynamics.

The first objective of this research is to design and implement a robotic patient-assistance framework using the UFACTORY xArm7 collaborative manipulator. The proposed framework enables the robotic arm to perform routine bedside assistance tasks through an integrated architecture that combines robotic manipulation, sensing, voice-based user interaction, and intelligent task execution. The system focuses on tasks such as pouring water, cleaning a designated area, delivering medicine, and taking temperature measurements. The overall objective is to demonstrate how collaborative robotic technology can provide reliable physical assistance while operating safely in environments shared with patients and healthcare professionals.

The second objective is to develop a voice-controlled human-robot interaction system that enables users to request assistance through natural spoken commands. A speech-to-text module is incorporated to capture the user's voice and convert spoken requests into textual information for further processing. This approach enables patients to interact with the robotic system without requiring physical access to a computer, touchscreen, or dedicated control interface, providing a more accessible method of interaction for users with limited mobility.

The third objective is to integrate a Large Language Model (LLM) into the voice-control framework for intelligent interpretation of user requests. The LLM is used to analyze recognized speech, identify the user's intended task, and map different natural-language expressions to predefined robotic actions. For example, expressions such as "Bring me some water," "Can I have some water?" and "I am thirsty" can be associated with the same water-assistance task. The LLM-based approach enables more flexible and intuitive communication by allowing users to express requests naturally rather than relying on a fixed set of predefined voice commands.

The fourth objective is to investigate autonomous object manipulation for routine patient-care applications. This work focuses on developing task-specific motion planning, grasping, and manipulation strategies that enable the xArm7 to retrieve, transport, position, and manipulate objects required for bedside assistance. The robotic system is intended to autonomously execute tasks identified through the voice-control framework, including water pouring, area cleaning, and medicine delivery. By automating predictable and repetitive manipulation activities, the proposed framework aims to reduce the need for direct human intervention while supporting routine patient assistance.

The fifth objective is to integrate non-contact physiological monitoring into the robotic assistance platform. In addition to performing physical manipulation tasks, the system is intended to support non-contact sensing technologies for acquiring physiological information, with particular emphasis on body temperature measurement. Integrating physiological monitoring with robotic manipulation allows the same platform to provide both physical assistance and basic patient observation. This combined capability establishes a foundation for future expansion toward additional non-contact physiological measurements and more comprehensive intelligent patient-monitoring systems.

II. MATERIALS AND METHODS

A. Hardware and Software Environment

The experimental platform employs a UFACTORY xArm7 collaborative robotic manipulator as the primary hardware for autonomous patient-assistance and vital-sign monitoring tasks. The xArm7 is a seven-degree-of-freedom (7-DoF) collaborative robot designed to provide kinematic redundancy, enabling flexible motion planning and obstacle avoidance in constrained environments while maintaining high positioning accuracy [32]. The robot offers a maximum payload capacity of 3.5 kg, a reach of 700 mm, and a repeatability of ± 0.1 mm, making it suitable for manipulation tasks that require precise positioning and safe human-robot interaction. Compared with conventional six-axis manipulators, the additional degree of freedom allows the robot to maintain multiple feasible arm configurations for the same end-effector pose, thereby improving dexterity and reducing the likelihood of singular configurations during operation [33].

The xArm7 is controlled through the UFACTORY controller using the official Python Software Development Kit (SDK), which provides high-level interfaces for joint-space motion, Cartesian trajectory planning, inverse kinematics, digital input/output (I/O), and gripper control. Communication between the host computer and the robot controller is established through a TCP/IP Ethernet connection, enabling seamless integration with external software developed in Python and Robot Operating System (ROS/ROS 2) environments. During system deployment, safety mechanisms including collision detection, self-collision avoidance, configurable safety boundaries, and emergency stop functions are enabled to ensure safe operation when the robot performs collaborative tasks in close proximity to users [34]. These hardware and software capabilities provide a flexible experimental platform for implementing intelligent robotic assistance systems in healthcare applications.

The Robot application was executed on an MSI gaming computer equipped with an Intel Core i7-13700F processor, an NVIDIA GeForce RTX 4070 graphics card with 12 GB of dedicated video memory, 32 GB DDR5 RAM, and a 1 TB NVMe solid-state drive running Windows 11 (64-bit) [35], [36], [37], [38]. These specifications exceed the recommended hardware requirements for real-time VR rendering, ensuring smooth operation and stable frame rates during the experimental evaluations.

We created a voice control component based on LLMs. The voice command was the input of a speech to text module, and LLMs created the intent classification, that directs the robot to perform certain tasks. The speech to text component was through python text to speech package and the LLMs was GPT package.

We created python algorithms to precisely control the robot arm's movements and interactions, forming the programming and control stage. During the programming and control stage, sophisticated control algorithms are developed using the Python programming language to enable precise and reliable operation of the robotic system. Python serves as the primary software development platform because of its extensive ecosystem of robotics libraries, numerical computation packages, and compatibility with the UFACTORY Software Development Kit (SDK). The developed software

is responsible for generating motion commands, performing inverse kinematics calculations, planning collision-free trajectories, and coordinating end-effector operations throughout each experimental task. Communication between the host computer and the robot controller is established through a TCP/IP Ethernet connection, allowing real-time transmission of motion commands and continuous monitoring of the robot's operating status.

B. System Design and Experimental Steps

The design goals are to create a robotic assistance framework, that is capable of Autonomous object manipulation with non-contact physiological monitoring. The software is capable of recognizing user input through voice recognition, mapping the task from the voice control, performing guided tasks, gathering data and recording of robot movement videos. Fig. 1 shows a diagram of the design objectives. The system design involves the following components. The Robotic Arm is located on a mobile table for easy manipulation, the connected computer serves as a hub for robot control and data aggregation. Python based tasks such as cleaning up, pouring water, delivering of medicine and vital sign measurement were established. On the user interface, the user can select tasks and record the robot actions during the tasks.

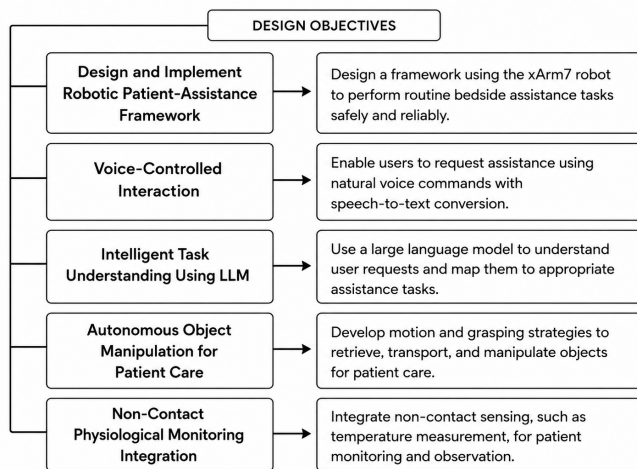


Fig. 1. Design objectives.

The software focused control package consisted the voice recognition component, the task identification component, and the robot task direction component. The voice recognition component utilized python text to speech package. The task identification component involves GPT package for intent mapping. The user interface built connections to direct robot for particular tasks.

The hardware related project methodology involves a comprehensive approach that comprises several key stages and assessment procedures. First, the UFACTORY Robot Arm is equipped with specialized end-effectors that can be designed for healthcare applications, and this involves the initial design and integration phase. Next, algorithms are developed using the Python programming language to precisely control the robot arm's movements and interactions, forming the programming

and control stage. To ensure reliability and safety, virtual environments are painstakingly created for thorough testing of the robot's actions before real-world deployment in the simulation and rigorous testing phase. Additionally, a user-friendly interface is carefully designed to provide healthcare professionals with seamless control over the robot arm's activities, constituting the user interface design phase. Throughout the process, data is systematically collected in both simulated and real healthcare scenarios, enabling in-depth analysis during the data acquisition and analysis stage. Finally, the performance of the automated assistant is evaluated using various metrics, including accuracy, operational efficiency, and we analyzed its impact on patient care, as part of the evaluation and impact assessment phase. Fig. 2 shows the system diagram involving both hardware and software.

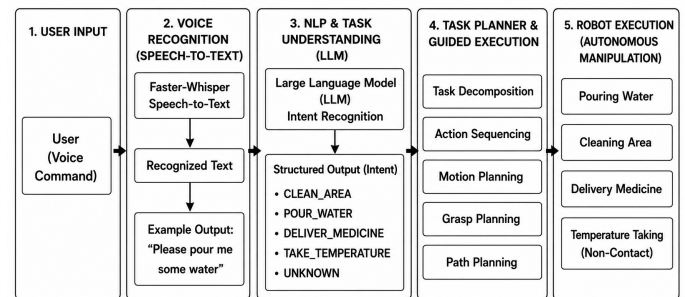


Fig. 2. System design diagram.

C. LLM-Based Natural Language Control Software Design

The speech-to-text component was designed to provide continuous voice recognition for natural human-robot interaction. Faster-Whisper was selected as the primary speech recognition framework because it provides a suitable balance between transcription accuracy and computational efficiency, making it appropriate for near-real-time robotic applications. The framework can be integrated into the Python-based control architecture to capture audio signals from a microphone, process the recorded speech, and convert the resulting audio into textual information. During operation, microphone input is continuously acquired and processed so that recognized speech can be displayed in real time and subsequently forwarded to the natural-language processing module. This continuous recognition approach allows users to provide successive verbal commands without manually restarting the speech-recognition process for each interaction. The transcribed text serves as the interface between the speech-recognition and Large Language Model components, where the recognized user request is further analyzed to determine the corresponding robotic task. If speech cannot be reliably recognized, the system can reject the input or request another command rather than initiating an uncertain robotic action. This implementation establishes a continuous voice-processing pipeline consisting of audio acquisition, speech recognition, text generation, intent interpretation, and robotic task selection, thereby supporting intuitive hands-free interaction with the patient-assistance system.

The LLM-based natural language control software is designed to provide an intuitive communication interface between the user and the robotic patient-assistance system.

Spoken commands are first captured through a microphone and converted into text using a speech-to-text module such as Faster-Whisper. The recognized text is then processed by a Large Language Model (LLM) to interpret the user's intent and associate different natural-language expressions with predefined robotic tasks. For example, commands such as "Bring me some water," "Can I have some water?" or "I am thirsty" can be interpreted as a water-assistance request. The LLM produces a structured task identifier corresponding to supported functions such as water pouring, area cleaning, medicine delivery, or temperature measurement. This task identifier is subsequently passed to the robot-control software, where predefined and validated motion routines are executed by the xArm7. By separating natural-language interpretation from low-level robotic control, the proposed software architecture provides flexible voice interaction while maintaining predictable and controlled robot behavior. Fig. 3 shows the diagram of voice recognition and LLM based task mapping code.

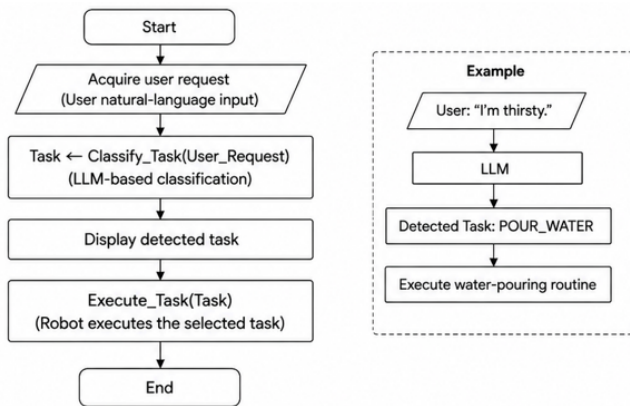


Fig. 3. Diagram of the process for the voice recognition and LLM-based task mapping.

D. Robot Installation and Initial Configuration

The installation of the UFACTORY xArm7 collaborative robotic system consists of several sequential steps to ensure safe and reliable operation. First, an appropriate workspace is defined to provide sufficient clearance for the robot's operating range and to prevent potential collisions with surrounding objects. The robotic arm is then securely mounted onto a stable work surface to ensure mechanical stability during operation. Next, the robotic arm is connected to the UFACTORY control box through the dedicated communication interface, followed by the connection of the control box to the power supply and the host computer using the required communication and power cables. After the hardware installation is completed, the system is powered on and initialized to verify proper communication between the robot controller and the host computer. Fig. 4 shows the connection of the robot arm with the computer that has the user interface, the robot control software component and robot task recording software component.

Following hardware installation, the robot undergoes an initial configuration and familiarization process using the *xArm Studio* software developed by UFACTORY. This software provides an integrated development environment for robot operation, programming, and system configuration. Users become

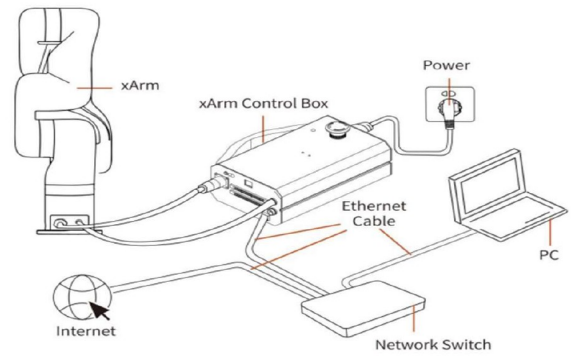


Fig. 4. Robot connection diagram.

familiar with the robot's hardware, control interface, operating modes, and workspace while verifying the robot's motion capability and communication status. Fig. 5 shows the central control unit of the system. Fig. 6 shows the XARM design interface. The software consists of five primary functional modules: *Live Control*, which enables manual manipulation of the robotic arm; *Blockly*, which supports graphical block-based programming; the *Python IDE*, which provides script-based robot programming and debugging; *Recording*, which allows robot trajectories to be recorded and replayed; and *Settings*, which manages system configuration, communication parameters, safety settings, and robot calibration. These integrated software tools facilitate rapid system deployment, robot programming, and experimental validation while providing a user-friendly interface for both novice and experienced users.



Fig. 5. Assisted robot Arm system - the central control unit.

The integration of the xArm presents several challenges, particularly in the realm of coordination. The mismatch between the coordinate systems of the xArm and the gripper introduces the potential for coordination issues during object manipulation. This misalignment can lead to inaccuracies and may result in failed attempts at grasping and pouring tasks. Additionally, the calibration process, essential for ensuring precise alignment between the two coordinate systems, proves to be a complex and time-consuming endeavor. Achieving perfect alignment demands meticulous adjustments, contributing to the overall intricacy of the setup.

Furthermore, an encountered challenge relates to IP configuration, where issues arose during power cycling. The IP

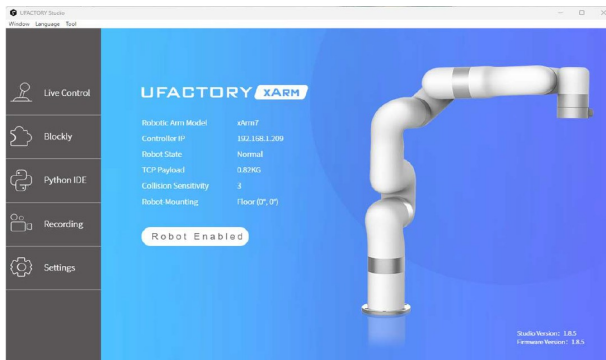


Fig. 6. The XARM studio design interface.

addresses of the computer and the control box were found to be on different network segments, posing a major hurdle. To address this, it became necessary to modify the computer's IP within the range of 192.168.1.1 - 192.168.1.255, with the tail number falling between 1 and 255, and ensuring it does not match the IP address of the control box. This adjustment was crucial for establishing a seamless network connection and overcoming the encountered IP configuration challenge.

E. Robot Control Software Design and Integration

We developed Python scripts that empower the UFACTORY robot arm to execute patient care tasks, ensuring precise and fluid movements. We created a software interface that bridges the gap between healthcare staff and the robot arm. This interface is designed to enable intuitive control, communication, and coordination, aligning seamlessly with the workflow of medical professionals. Additionally, the performance of the robot arm is recorded, with the goal for validation of calibration, and task completion testing.

Live Control offers the capability to manipulate the xArm's position and adjust its posture using XYZ coordinates within the base coordinate system for the arm and the tool coordinate system for the gripper. Fig. 7 shows the interface of the Live Control of the Assisted Robot Arm.

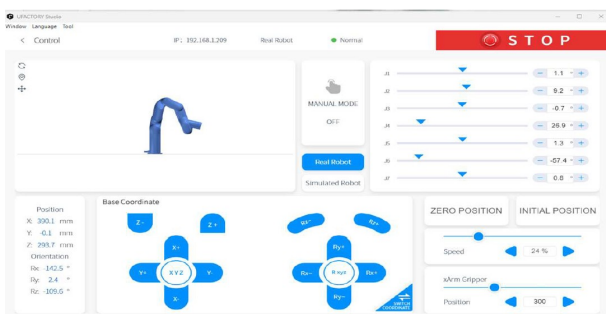


Fig. 7. Interface of live control of the assisted robot Arm.

Blockly, a graphical programming tool, simplifies programming tasks by allowing users to drag and drop code blocks to control the robotic arm, input, and output. Fig. 8 shows the Blockly interface for experimenting kinematics parameters. The diagram illustrates the block-based programming workflow used to control the UFACTORY xArm7 robotic joints

and associated end-effector functions. The process begins with the Move Servo (Joint Motion) command, which specifies the target joint, desired angular position, starting angle, motion duration, speed, and execution mode. These parameters determine how an individual robot joint moves from its initial configuration to a defined target position. The workflow then incorporates the Set Analog Output command, which can be used to regulate auxiliary devices such as the gripper by assigning an output pin and control value. Subsequent joint movements can be generated by repeating the servo command with updated target parameters, allowing a sequence of coordinated robot poses to be constructed.

The final stage uses the Set End I/O Pin (Digital Output) command to control binary end-effector functions, such as opening or closing the gripper. The supporting reference tables summarize the principal command parameters and identify the seven joints of the xArm7, from the base joint to the distal wrist joints. Together, these programming elements provide a structured approach for defining joint trajectories, controlling end-effector operations, and coordinating sequential robot actions. The block-based interface therefore enables robot motion to be programmed without requiring extensive low-level coding, while still providing control over position, speed, timing, and I/O states.



Fig. 8. Blockly interface for experimenting kinematics parameters.

The Python IDE facilitates direct utilization of the Python SDK API for creating programs that instruct the robotic arm to perform particular tasks. After experimenting with Blockly for key parameters, we program each task as a Python program. Fig. 9 shows an example Python pseudocode for the task of pouring water.

III. RESULTS

A. Results of the Natural Language Based LLM Enabled Task Mapping

Fig. 10 shows an example display of python code execution for the natural language based LLM enabled task mapping. The output demonstrates the successful operation of the LLM based natural language task classification module. The system receives transcribed user commands and interprets their semantic meaning to determine the corresponding predefined robotic task. For example, the statement "I'm thirsty" is classified as POUR_WATER, demonstrating that the system can infer the user's intention even when the requested action is not explicitly stated. Similarly, "Please clean the table" is mapped to CLEAN_AREA, while "Check my temperature" is classified as TAKE_TEMPERATURE. These results demonstrate that the LLM can associate different natural-language expressions with

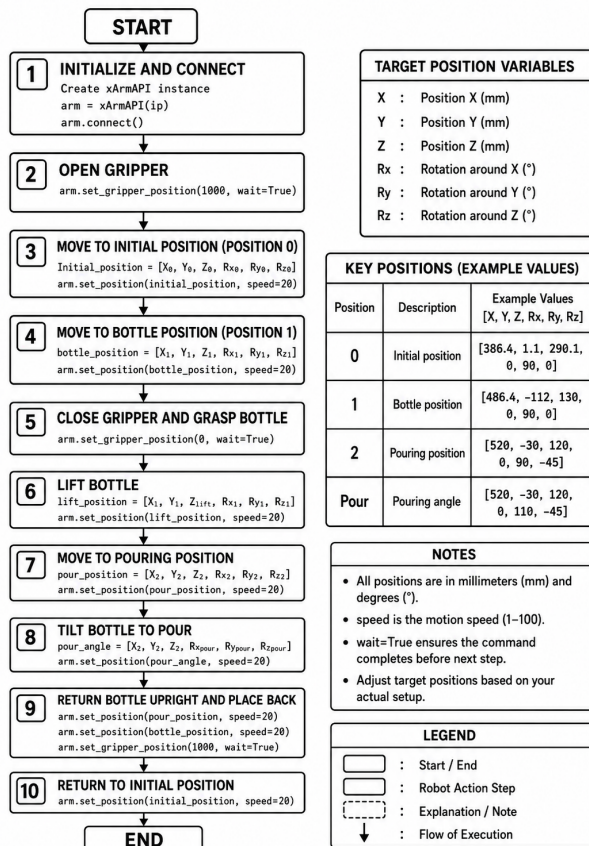


Fig. 9. Pseudo code and illustrations of the assisted robot Arm for pouring water.

standardized task identifiers, which can subsequently be passed to the robot-control module to initiate the appropriate predefined task routine.

```
You said: I'm thirsty
Detected task: POUR_WATER

You said: Please clean the table
Detected task: CLEAN_AREA

You said: Check my temperature
Detected task: TAKE_TEMPERATURE
```

Fig. 10. Python output for voice recognition and LLM based task mapping.

B. Assisted Robotic Arm Performance for Example Tasks and Use Cases

Empirical evidence forms the cornerstone of this research's contributions. A comprehensive account of the data was collected during task execution, including completion times, accuracy rates, and any observed safety considerations, was provided. These results were contextualized within the research's objectives and contribute to the overall evaluation of the robot arm's efficacy and feasibility in healthcare scenarios. The different tasks were divided into cleaning area, pouring Water, delivering medicine, taking body temperature.

1) *Remove medical contaminated materials:* The deployment of the UFACTORY xArm7 robotic arm and gripper for the removal of medical contaminated materials in healthcare settings had made significant strides, leveraging advanced robotic technology to enhance efficiency and safety. The deployment process began with the installation of the xArm Python Library, facilitating seamless communication with the robotic arm. Following this, the necessary modules were imported to enable comprehensive control and programming capabilities. An instance of the xArm API class was created, specifying the IP address or port of the xArm7 to establish a connection. Through the "connect" method, the xArm seamlessly linked to the designated system. The deployment strategy incorporated a loop to repeat the material removal task multiple times, allowing for systematic and repetitive operations. Positions of the xArm were dynamically adjusted within the loop to navigate the robot precisely to the targeted contaminated materials. Upon completing the task iterations, the xArm disconnected from the system, marking the successful execution of the deployment plan. This progress highlighted the successful integration of advanced robotic solutions for the efficient and controlled removal of medical contaminated materials in healthcare settings.

2) *Deliver water / pouring water into the cup:* In a healthcare context, implementing the precise and controlled pouring of water from a bottle into a cup using the UFACTORY xArm7 robotic arm equipped with a gripper offered a range of potential benefits. This application was particularly useful in scenarios where patients have difficulty performing such tasks independently due to physical limitations or health conditions. The robotic arm's precision allowed for accurate positioning over the cup, ensuring a controlled and spill-free pouring process. The gripper's dexterity enabled the arm to securely hold and manipulate the water bottle, providing a level of assistance that could contribute to patients' autonomy and well-being. By automating simple yet essential tasks like pouring water, the UFACTORY xArm7 and gripper combination enhanced the efficiency of healthcare facilities, allowing healthcare professionals to focus on more complex aspects of patient care while simultaneously promoting patients' independence and comfort. Developing a Python code implementation for pouring water from a bottle into a cup using the UFACTORY xArm7 robotic arm and gripper in a healthcare setting involved integrating precision and control to assist patients with limited mobility. The code initiated the robotic arm's movement to grasp the water bottle securely with the gripper, navigated to a predetermined position over the cup, and gradually tilted the bottle to pour water accurately. Implementing safety measures was crucial, ensuring a controlled pouring process that minimizes the risk of spillage. Additionally, the code could incorporate feedback mechanisms to adapt to varying bottle sizes or environmental factors. This automation aimed to empower patients in healthcare scenarios, promoting independence and alleviating the burden on healthcare professionals for routine caregiving tasks, enhancing the overall quality of patient care.

3) *Deliver food / deliver medicine:* Creating a Python code implementation for delivering food or medicine to patients using the UFACTORY xArm7 robotic arm and gripper introduced a transformational aspect to healthcare assistance. The code was designed to facilitate the robotic arm's precise movement to pick up a designated package containing

either food or medicine with the gripper. It then navigated through predefined paths to reach the patient's location and gently release the package. Implementing safety protocols was paramount to ensure the robotic arm operates in a controlled manner, avoiding any harm to patients or surroundings. The code also incorporated obstacle detection and avoidance mechanisms to navigate through dynamic environments. By automating the delivery process, this code aimed to enhance operational efficiency within healthcare facilities, allowing healthcare professionals to focus on more intricate aspects of patient care, while simultaneously providing timely and convenient services to patients in need.

4) *Take temperature*: Implementing a Python code for the UFACTORY xArm7 robotic arm and gripper to take the temperature of patients in a healthcare setting introduced a sophisticated level of automation. The code instructed the robotic arm to grasp a non-contact infrared thermometer using the gripper, ensuring a secure hold. The arm then carefully approached the patient, positioned the thermometer at the appropriate distance, and triggered the temperature measurement. To enhance the code's functionality, it incorporated features such as patient identification, precise positioning, and data logging for accurate record-keeping. Safety measures, including controlled movement and obstacle detection, were crucial to guarantee the well-being of patients during the temperature-checking process. By automating temperature measurements, this code contributed to the efficiency of healthcare operations, reduced the risk of cross-contamination, and allowed healthcare professionals to allocate more time to critical aspects of patient care.

C. Experimental Performances of Assisted Robotic Arm

The experimental performances of the assisted robot arm were taken as videos. Screen captures of the videos were gathered to show the performance of the assisted arm. Fig. 11 shows assisted robot arm cleaning area, pouring Water, delivering medicine and taking body temperature. Fig. 12 shows the example process of assisted robot arm pouring water.



Fig. 11. Assisted robot Arm cleaning area, pouring water, delivering medicine, taking body temperature.

D. Recorded Robot Movement in 3D modeling Environment of the Assisted Robotic Arm

The UFACTORY xArm Studio software provides a three-dimensional modeling environment for recording, visualizing, and replaying robotic movements. During task execution, the positions and orientations of the robotic arm are continuously recorded and displayed using a real-time 3D model that mirrors the physical robot. An example is shown in Fig. 13. This environment allows users to verify joint trajectories, monitor

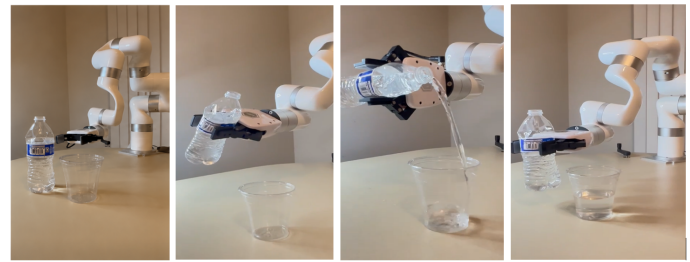


Fig. 12. Assisted robot Arm completing tasks of pouring water.

end-effector motion, and evaluate the sequence of programmed operations before deployment. The recording function also enables previously executed motions to be replayed for debugging, trajectory refinement, and performance validation. By providing a synchronized digital representation of the physical robotic system, the 3D modeling environment facilitates efficient program development and improves the reliability of healthcare assistance tasks.

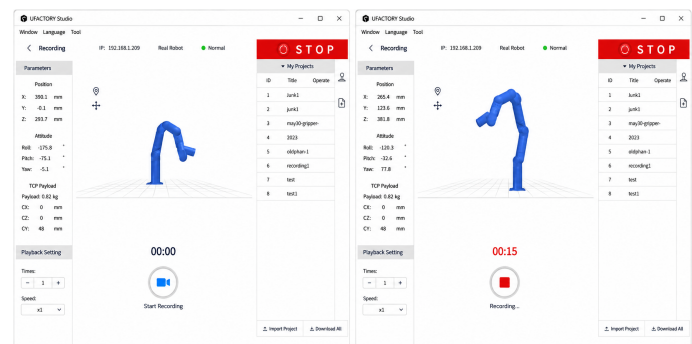


Fig. 13. Recorded robot movement in 3D modeling environment.

IV. SYSTEM VALIDATION OF ASSISTED ROBOTIC ARM PERFORMANCE FOR EXAMPLE TASKS AND USE CASES

A total of 100 different voice commands related to the supported robotic healthcare tasks were evaluated to assess the performance of the LLM-based task classification system. The test commands represented different natural-language expressions associated with four predefined task categories: POUR_WATER, CLEAN_AREA, DELIVER_MEDICINE, and TAKE_TEMPERATURE. Rather than requiring users to speak an exact predefined phrase, the experiment included variations in wording and play the wording with audio to evaluate whether the LLM could identify the intended robotic action from the semantic meaning of each command. Of the 100 commands, 97 were mapped to the intended task category, resulting in an overall classification accuracy of 97%. The three classification errors were associated with commands containing ambiguous or overlapping semantic information. For example, an expression such as "pouring medicine" can contain information related to both the POUR_WATER and DELIVER_MEDICINE categories. These results indicate that the LLM-based approach can provide flexible natural-language control while also demonstrating the need for ambiguity detection and confirmation mechanisms before physical robot execution.

Table I presents the confusion-matrix representation of the classification results. The diagonal elements indicate correctly classified commands, whereas the off-diagonal elements represent classification errors. In this example, 25 commands were evaluated for each task category, producing 97 correct classifications and three misclassifications.

TABLE I. CONFUSION MATRIX FOR LLM-BASED TASK CLASSIFICATION

Actual Task	Predicted Task			
	Water	Clean	Medicine	Temp.
Water	24	0	1	0
Clean	0	25	0	0
Medicine	1	0	24	0
Temperature	0	0	1	24

The three off-diagonal entries represent the three classification errors. The bold diagonal values make the correct classifications particularly easy to identify.

The confusion matrix shows that most predictions are concentrated along the main diagonal, demonstrating strong agreement between the intended task and the task selected by the LLM. The remaining errors illustrate an important limitation of natural-language robotic control: semantically similar or ambiguous expressions can occasionally be assigned to an incorrect action. This issue is particularly important in healthcare-oriented applications because an incorrect classification could initiate an unintended robotic operation. Future versions of the system can therefore incorporate an UNKNOWN or AMBIGUOUS category, confidence thresholds, and a confirmation step in which the system asks the user to verify the interpreted task before executing potentially safety-sensitive actions. These mechanisms could further improve the reliability of the natural-language interface while preserving the flexibility provided by LLM-based intent interpretation.

The robotic healthcare assistant was validated using repeated task-based experiments. Four representative functions were evaluated: removal of contaminated materials, water pouring, medicine delivery, and body-temperature measurement. Each task was performed ten times under the same experimental conditions. Task success, execution time, end-effector positioning error, repeatability, and operational safety were recorded for every trial.

For motion validation, the commanded Cartesian coordinates were compared with the measured end-effector coordinates. The three-dimensional positioning error was calculated as:

$$E = \sqrt{(X_m - X_t)^2 + (Y_m - Y_t)^2 + (Z_m - Z_t)^2}, \quad (1)$$

where, (X_t, Y_t, Z_t) represents the commanded target position and (X_m, Y_m, Z_m) represents the measured position. Repeatability was evaluated using the standard deviation of the measured coordinates across repeated trials. A task was considered successfully validated when at least 90% of the trials were completed correctly, the average positioning error remained below 5 mm, and no critical collision, uncontrolled motion, or object-dropping event occurred. The validation

results were then compared across the four healthcare tasks to evaluate the reliability of the complete robotic system. The validation criteria for the robot healthcare system is as Table II.

Table III shows the validation details for pouring water. For the pouring-water function specifically, validation test the full sequence from opening the gripper through returning the bottle. The bottle is placed at a fixed starting location, and the cup remains at a known target location. Each trial determine whether the robot successfully grasps the bottle, lifts it without slipping, reaches the pouring position, tilts the bottle, returns it upright, replaces it, and releases it correctly.

TABLE II. VALIDATION CRITERIA FOR THE ROBOTIC HEALTHCARE SYSTEM

Metric	Measurement	Acceptance Criterion
Task Success	Successful trials / total trials	$\geq 90\%$
Position Error	Euclidean distance between target and measured XYZ	≤ 5 mm
Repeatability	Standard deviation of repeated XYZ measurements	≤ 3 mm
Execution Time	Time required to complete each task	Stable across trials
Command Response	Correct execution after user/software command	$\geq 90\%$
Safety	Collision, dropped object, or uncontrolled motion	0 critical events

TABLE III. VALIDATION OF THE ROBOTIC WATER-POURING SEQUENCE

Step	Operation	Expected Result	Criterion	Status
1	Open gripper	Gripper opens	Correct position	Pass
2	Move to bottle	Reaches bottle	Error ≤ 5 mm	Pass
3	Grasp bottle	Bottle secured	No slipping	Pass
4	Lift bottle	Reaches safe height	Error ≤ 5 mm	Pass
5	Move above cup	Reaches pour point	Error ≤ 5 mm	Pass
6	Tilt bottle	Water poured	Correct rotation	Pass
7	Return upright	Pouring stops	No continued flow	Pass
8	Replace bottle	Returns to target	Error ≤ 5 mm	Pass
9	Release bottle	Bottle stable	No tipping	Pass
10	Return arm	Reaches home	Error ≤ 5 mm	Pass

TABLE IV. EXPERIMENTAL VALIDATION RESULTS

Task	Trials	Success	Rate (%)	Time (s)	Error (mm)	Result
Remove Material	10	10	100	10.4	1.42	Pass
Pour Water	10	9	90	10.8	1.61	Pass
Deliver Medicine	10	10	100	9.7	1.28	Pass
Measure Temperature	10	10	100	10.1	1.35	Pass
Overall	40	39	97.5	10.25	1.42	Pass

Table IV presents the experimental validation results of the robotic healthcare assistance system for four representative tasks: removing contaminated material, pouring water, delivering medicine, and measuring body temperature. Each task was repeated ten times to examine the reliability of the robotic arm under consistent experimental conditions. The

system successfully completed 39 of the 40 trials, resulting in an overall task success rate of 97.5%. Three tasks achieved a 100% success rate, while the water-pouring task achieved 90%, indicating one unsuccessful trial. The lower success rate for water pouring can be attributed to the greater manipulation complexity of this task, which requires the robot to grasp the bottle, lift it, move toward the target, control the wrist orientation during pouring, and return the bottle to its original position.

The execution time remained relatively consistent among the four tasks, ranging from 9.7 s for medicine delivery to 10.8 s for water pouring. The overall mean execution time was 10.25 s. Water pouring required the longest time because it involved additional movements for grasping, lifting, tilting, and returning the bottle. In comparison, medicine delivery produced the shortest execution time. These results suggest that the robotic system can complete the selected point-of-care activities within a relatively consistent time range despite differences in task requirements.

Positioning performance was evaluated using the mean error between the target and measured end-effector positions. As shown in Table IV, the mean errors ranged from 1.28 mm to 1.61 mm, with an overall mean of 1.42 mm. Medicine delivery produced the lowest positioning error of 1.28 mm, whereas water pouring produced the highest error of 1.61 mm. The increased error during pouring may result from the additional changes in arm position and wrist orientation required to manipulate the bottle. Nevertheless, all measured errors remained below the predefined 5 mm acceptance threshold. The combination of a 97% overall success rate, an average execution time of 10.25 s, and a mean positioning error of 1.42 mm indicates that the developed robotic arm system can perform the tested healthcare assistance tasks with good repeatability and operational reliability. These results support the feasibility of the proposed system for further development as an intelligent robotic assistant for routine point-of-care applications.

A. Discussion of Errors and Potential Improvements

The experimental results demonstrate that the proposed system can successfully integrate voice recognition, LLM-based intent classification, and robotic task execution; however, several sources of error were observed. For the natural-language control component, the LLM-based task classification achieved an accuracy of 97%, indicating that a small proportion of the tested commands were incorrectly classified. One important source of error was ambiguity in the user's spoken request. For example, an expression such as "pouring medicine" contains semantic information associated with both the POUR_WATER and DELIVER_MEDICINE classes. Consequently, the LLM may assign an incorrect task when the input contains overlapping terminology or does not clearly describe the intended action. Speech-to-text errors can further contribute to incorrect classifications because inaccurate transcription changes the textual input provided to the LLM.

Several approaches can be used to improve the reliability of the natural-language control component. First, an UNKNOWN class can be incorporated so that ambiguous or low-confidence commands do not immediately initiate robotic motion. Instead,

the system can request clarification from the user, such as asking whether the user intends to pour water or receive medicine. Second, task-specific prompts and additional representative command examples can be provided to the LLM to improve intent classification. The system can also require confirmation before executing safety-sensitive activities, particularly medicine-related tasks. These modifications would reduce the probability that an incorrectly interpreted command results in an unintended robotic action.

Errors can also occur during physical robotic execution. Although predefined Cartesian coordinates allow the xArm7 to repeatedly approach known target locations, small positioning differences can result from calibration uncertainty, coordinate-system alignment, object-placement variation, gripper positioning, and mechanical repeatability. Manipulation tasks such as grasping a bottle and pouring water are particularly sensitive because a small displacement of the bottle or cup can affect grasping and pouring performance. Differences between commanded and measured end-effector coordinates therefore provide an important quantitative indication of robotic positioning error.

Currently the level of intelligence of automation is at the basic level where the motion were predefined without sensor input to enhance the level of intelligence. Future improvements should incorporate additional sensor feedback rather than relying primarily on predefined positions and motion sequences. A camera-based vision system could identify the real-time locations of bottles, cups, medicine containers, and other objects before manipulation. Visual feedback could then be used to dynamically modify the target coordinates instead of assuming that objects remain at exactly the same locations. Force or tactile sensing at the gripper could additionally determine whether an object has been successfully grasped and detect unexpected contact. Combining visual perception, force feedback, collision detection, and closed-loop trajectory correction would enable the robotic system to adapt to variations in the environment and improve manipulation reliability.

Further improvements also include adding more trials for the validation. For example, for the validation of the LLMs, we didn't test on the execution time and potential delays although through observation, those were not the major potential problems for the system to function. Moreover, for the assisted robot arm's performance, the 97.5 percent overall success rate and a 1.42 mm mean positioning error, rest on only ten trials per task, so the work can be toward reporting trial-level variability or increase the sample size. More comparisons could be done to compare with previous studies in the same scope.

Overall, the observed errors highlight the importance of validating both the intelligent interaction layer and the physical robotic layer. The LLM provides flexibility in understanding natural-language requests, while predefined robot routines provide controlled task execution; nevertheless, uncertainty can be introduced at the speech-recognition, intent-classification, coordinate-calibration, and manipulation stages. Future development should therefore emphasize confidence-based intent classification, user confirmation for ambiguous commands, improved calibration, real-time object detection, sensor-based feedback, and closed-loop robotic control. These improvements could increase the robustness of the proposed system

and provide a stronger foundation for future evaluation in more realistic healthcare environments.

V. DISCUSSIONS

The findings of this study demonstrate the feasibility of employing a collaborative robotic manipulator as an intelligent assistant for healthcare applications. The proposed system successfully integrates robotic manipulation, autonomous task execution, and software-based control into a unified platform capable of supporting routine patient-care activities. By automating repetitive bedside tasks, such as object delivery and physiological monitoring, the robotic assistant has the potential to reduce the physical workload of healthcare professionals while allowing them to devote greater attention to clinical decision-making and direct patient interaction. The integration of collaborative robotics into healthcare environments therefore represents a promising approach for improving operational efficiency without replacing the essential role of medical personnel.

One of the primary contributions of this work is the development of a modular software architecture that combines robot motion control, task planning, user interaction, and potential sensor integration using Python. The modular design enables individual system components to be independently updated or expanded, facilitating future integration of additional sensing devices and artificial intelligence techniques.

The practical deployment of collaborative robotic assistants requires consideration of economic and operational factors. As collaborative robot technology continues to mature and hardware costs decrease, robotic assistants may become increasingly cost-effective, particularly for performing predictable and repetitive tasks that consume substantial nursing resources. Improvements in usability, interoperability with hospital information systems, and standardized deployment procedures are planned for future work.

VI. CONCLUSIONS AND FUTURE WORK

This study developed and evaluated an intelligent robotic patient-assistance framework that integrates collaborative robotic manipulation, voice-based natural-language interaction, and non-contact physiological monitoring. The UFACTORY xArm7 collaborative manipulator was used as the primary robotic platform to perform representative patient-care tasks, including cleaning a designated area, pouring water, delivering medicine, and taking body temperature. The proposed framework demonstrates how a collaborative robotic arm can be programmed to provide physical assistance for predictable and repetitive bedside activities while maintaining a software architecture that can be expanded to accommodate additional healthcare-related functions.

A significant contribution of this research is the integration of natural-language voice control with robotic task execution. The developed software uses continuous speech-to-text processing to convert spoken user requests into textual information, which is subsequently analyzed by a Large Language Model to determine the intended robotic task. This approach enables users to communicate with the robotic assistant using natural expressions rather than memorizing a fixed set of

commands. Evaluation demonstrates the potential of LLM-based intent recognition for intuitive human-robot interaction. However, ambiguous expressions, such as commands containing terminology associated with multiple tasks, can result in incorrect classifications.

The integration of non-contact physiological monitoring further extends the functionality of the robotic platform beyond conventional object manipulation. In this study, body-temperature measurement was incorporated as a representative monitoring task in which the robotic arm positions a non-contact infrared thermometer for patient measurement. Combining physical assistance, physiological sensing, and natural-language interaction within the same robotic framework provides a foundation for developing more comprehensive patient-support systems. Future implementations could extend this architecture to additional non-contact sensors and more advanced perception capabilities while preserving appropriate clinical supervision.

Overall, the findings demonstrate the feasibility of combining collaborative robotics, Python-based robot control, LLM-enabled natural-language interaction, and physiological sensing within a unified healthcare-assistance framework. The proposed system is not intended to replace nurses or other healthcare professionals; instead, it provides a technological approach for supporting selected repetitive and predictable activities that can consume caregiver time. Future research should focus on improving recognition of ambiguous commands, integrating environmental perception and obstacle avoidance, expanding physiological sensing capabilities, and conducting evaluations in more realistic patient-care environments. Further development of these capabilities could enable robotic assistants to provide more adaptive and accessible support while allowing healthcare professionals to concentrate on clinical assessment, decision-making, and direct patient interaction.

Beyond healthcare assistance, the proposed collaborative robotic platform has potential to support individuals with upper-limb impairments in performing activities of daily living. A collaborative robotic arm positioned beside a bed or wheelchair can assist with tasks such as retrieving personal items, delivering food or medication, and providing drinking assistance through intuitive user interfaces, including touch or voice commands. By combining autonomous manipulation with intelligent motion planning, the robotic system can improve user independence, reduce reliance on caregivers, and enhance quality of life. Future research may incorporate advanced perception and artificial intelligence to provide more adaptive, personalized assistance in home and rehabilitation environments.

REFERENCES

- [1] World Health Organization. *State of the World's Nursing 2025: Investing in Education, Jobs, Leadership and Service Delivery*; World Health Organization: Geneva, Switzerland, 2025.
- [2] L. Z. Li, L. X. Yang, L. J. Liu, and F. F. Wu, "Nurse burnout and patient safety, satisfaction, and quality of care: A systematic review and meta-analysis," *JAMA Network Open*, vol. 7, no. 11, p. e2443059, 2024. doi: 10.1001/jamanetworkopen.2024.43059.
- [3] "Contributing factors to missed nursing care: A systematic review with clinical implications for practice and patient safety," *Applied Nursing Research*, vol. 85, p. 152002, 2025. doi: 10.1016/j.apnr.2025.152002.

- [4] X. Meng, X. Yan, K. Zhang, *et al.*, "The application of large language models in medicine: A scoping review," *iScience*, vol. 27, no. 5, Art. no. 109713, 2024. doi: 10.1016/j.isci.2024.109713.
- [5] F. Busch, L. Hoffmann, C. Rueger, *et al.*, "Current applications and challenges in large language models for patient care: A systematic review," *Communications Medicine*, vol. 5, Art. no. 26, 2025.
- [6] Ball, J.E.; Bruyneel, L.; Aiken, L.H.; Sermeus, W.; Sloane, D.M.; Rafferty, A.M.; Lindqvist, R.; Tishelman, C.; Griffiths, P. Post-operative mortality, missed care and nurse staffing in nine countries: A cross-sectional study. *Int. J. Nurs. Stud.* **2018**, *78*, 10–15.
- [7] Ohneberg, C.; Stöbich, N.; Warmbein, A.; Rathgeber, I.; Mehler-Klamt, A.C.; Fischer, U.; Eberl, I. Assistive robotic systems in nursing care: A scoping review. *BMC Nurs.* **2023**, *22*, 72.
- [8] Bemelmans, R.; Gelderblom, G.J.; Jonker, P.; de Witte, L. Socially assistive robots in elderly care: A systematic review into effects and effectiveness. *J. Am. Med. Dir. Assoc.* **2012**, *13*, 114–120.e1.
- [9] Broadbent, E.; Stafford, R.; MacDonald, B. Acceptance of healthcare robots for the older population: Review and future directions. *Int. J. Soc. Robot.* **2009**, *1*, 319–330.
- [10] Papadopoulos, I.; Koulouglioti, C.; Ali, S. Views of nurses and other health and social care workers on the use of assistive humanoid and animal-like robots in health and social care: A scoping review. *Contemp. Nurse* **2020**, *56*, 102–119.
- [11] Feil-Seifer, D.; Mataric, M.J. Defining socially assistive robotics. In *Proceedings of the IEEE International Conference on Rehabilitation Robotics*; Chicago, IL, USA, 2005; pp. 465–468.
- [12] Goodrich, M.A.; Schultz, A.C. Human–robot interaction: A survey. *Found. Trends Hum.-Comput. Interact.* **2008**, *1*, 203–275.
- [13] Bhattacharjee, T.; Gordon, E.K.; Srinivasa, S.S. Assistive robot manipulation for activities of daily living. *Annu. Rev. Control Robot. Auton. Syst.* **2020**, *3*, 1–27.
- [14] Park, D.; Hoshi, Y.; Mahajan, H.P.; Kim, H.K.; Erickson, Z.; Rogers, W.A.; Kemp, C.C. Active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned. *Robot. Auton. Syst.* **2020**, *124*, 103344.
- [15] Siegel, J.D.; Rhinehart, E.; Jackson, M.; Chiarello, L. *2007 Guideline for Isolation Precautions: Preventing Transmission of Infectious Agents in Healthcare Settings*; Centers for Disease Control and Prevention: Atlanta, GA, USA, 2007.
- [16] Nelson, A.; Baptiste, A.S. Evidence-based practices for safe patient handling and movement. *Online J. Issues Nurs.* **2006**, *11*, 3.
- [17] Davis, K.G.; Kotowski, S.E. Prevalence of musculoskeletal disorders for nurses in hospitals, long-term care facilities, and home health care: A comprehensive review. *Hum. Factors* **2015**, *57*, 754–792.
- [18] Barr, P.; Zhang, Q.; Rostami, M.R. Protecting healthcare workers during infectious disease outbreaks: Occupational exposure and control strategies. *J. Hosp. Infect.* **2020**, *105*, 608–615.
- [19] Mutlu, B.; Roy, N.; Forlizzi, J.; Hodgins, J. Robots in healthcare: Designing collaborative robotic assistants for clinical environments. *Annu. Rev. Biomed. Eng.* **2020**, *22*, 339–364.
- [20] Dario, P.; Guglielmelli, E.; Allotta, B.; Carrozza, M.C. Robotics for Medical Applications. *IEEE Robot. Autom. Mag.* **2000**, *7*, 44–56.
- [21] Kragic, D.; Christensen, H.I. Survey on Visual Servoing for Manipulation. *Comput. Vis. Image Underst.* **2009**, *113*, 1–16.
- [22] Tsui, K.M.; Yanco, H.A.; Cramer, H.; Kemper, N. Using Autonomous Robots to Support Healthcare Workers in Hospitals. In *Proceedings of the ACM/IEEE International Conference on Human–Robot Interaction (HRI)*; Lausanne, Switzerland, 2011; pp. 209–216.
- [23] Javdani, S.; Bagnell, J.A.; Srinivasa, S.S. Shared Autonomy via Hindsight Optimization. *Int. J. Robot. Res.* **2018**, *37*, 717–742.
- [24] Kim, B.; Pineau, J. Socially Adaptive Path Planning in Human Environments Using Inverse Reinforcement Learning. *Int. J. Soc. Robot.* **2016**, *8*, 51–66.
- [25] Maheu, V.; Frappier, J.; Archambault, P.; Routhier, F. Evaluation of the JACO Robotic Arm: Clinico-Economic Study for People with Upper-Extremity Disabilities. In *Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR)*; Zurich, Switzerland, 2011.
- [26] Heerink, M.; Kröse, B.; Evers, V.; Wielinga, B. Assessing Acceptance of Assistive Social Robots by Older Adults: The Almere Model. *Int. J. Soc. Robot.* **2010**, *2*, 361–375.
- [27] Beer, J.M.; Smarr, C.A.; Chen, T.L.; Prakash, A.; Mitzner, T.L.; Kemp, C.C.; Rogers, W.A. The Domesticated Robot: Design Guidelines for Assisting Older Adults to Age in Place. In *Proceedings of the ACM/IEEE International Conference on Human–Robot Interaction (HRI)*; Boston, MA, USA, 2012; pp. 335–342.
- [28] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, "Large language models in medicine," *Nature Medicine*, vol. 29, pp. 1930–1940, 2023. doi: 10.1038/s41591-023-02448-8.
- [29] J. N. Kather, D. Ferber, I. C. Wiest, S. Gilbert, and D. Truhn, "Large language models could make natural language again the universal interface of healthcare," *Nature Medicine*, vol. 30, pp. 2708–2710, 2024. doi: 10.1038/s41591-024-03199-w.
- [30] D. Wang and S. Zhang, "Large language models in medical and healthcare fields: Applications, advances, and challenges," *Artificial Intelligence Review*, vol. 57, Art. no. 299, 2024. doi: 10.1007/s10462-024-10921-0.
- [31] K. Singhal, S. Azizi, T. Tu, *et al.*, "Large language models encode clinical knowledge," *Nature*, vol. 620, pp. 172–180, 2023.
- [32] UFACTORY, *xArm User Manual*. Shenzhen, China: UFACTORY Robotics, 2025.
- [33] UFACTORY, *xArm Studio User Guide*. Shenzhen, China: UFACTORY Robotics, 2025.
- [34] UFACTORY, *xArm Developer Documentation*. Shenzhen, China: UFACTORY Robotics, 2025. [Online]. Available: <https://docs.xarm.ufactory.cc>
- [35] Intel Corporation, "Intel Core i7-13700F processor specifications," [Online]. Available: <https://www.intel.com/content/www/us/en/products/sku/230490/intel-core-i713700f-processor-30m-cache-up-to-5-20-ghz/specifications.html>. [Accessed: Aug. 3, 2026].
- [36] NVIDIA Corporation, "GeForce RTX 4070 specifications," [Online]. Available: <https://www.nvidia.com/en-us/geforce/graphics-cards/40-series/rtx-4070/>. [Accessed: Aug. 3, 2026].
- [37] Micro-Star International (MSI), "MSI gaming desktop products," [Online]. Available: <https://www.msi.com/Desktop>. [Accessed: Aug. 3, 2026].
- [38] Microsoft, "Windows 11 specifications and system requirements," [Online]. Available: <https://www.microsoft.com/windows/windows-11-specifications>. [Accessed: Aug. 3, 2026].