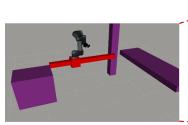
Gesture-Based Robot Control Integrating Mm-wave Radar and Behavior Trees

Yuqing Song* yuqing.song@aalto.fi Aalto University Espoo, Finland

Sanaz Kianoush sanaz.kianoush@cnr.it National Research Council of Italy Milan, Italy Cesare Tonola*
cesare.tonola@cnr.it
National Research Council of Italy
Milan, Italy

Nicola Pedrocchi nicola.pedrocchi@stiima.cnr.it National Research Council of Italy Milan, Italy Stefano Savazzi stefano.savazzi@cnr.it National Research Council of Italy Milan, Italy

> Stephan Sigg stephan.sigg@aalto.fi Aalto University Espoo, Finland



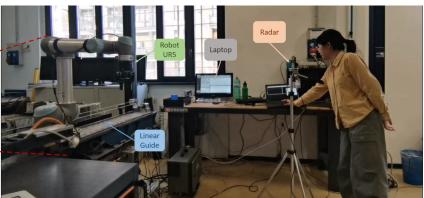


Figure 1: Experimental site of millimeter-wave radar-based gesture-controlled robotic arm. The inset on the left shows the digital twin of the robotic arm and linear guide used for on-site demonstration.

Abstract

As robots become increasingly prevalent in both homes and industrial settings, the demand for intuitive and efficient human-machine interaction continues to rise. Gesture recognition offers an intuitive control method that does not require physical contact with devices and can be implemented using various sensing technologies. Wireless solutions are particularly flexible and minimally invasive. While camera-based vision systems are commonly used, they often raise privacy concerns and can struggle in complex or poorly lit environments. In contrast, radar sensing preserves privacy, is robust to occlusions and lighting, and provides rich spatial data such as distance, relative velocity, and angle. We present a gesturecontrolled robotic arm using mm-wave radar for reliable, contactless motion recognition. Nine gestures are recognized and mapped to real-time commands with precision. Case studies are conducted to demonstrate the system practicality, performance and reliability for gesture-based robotic manipulation. Unlike prior work that treats gesture recognition and robotic control separately, our system unifies both into a real-time pipeline for seamless, contactless human-robot interaction.

Keywords

Robotic Arm Control, Human-Machine Interaction, Gesture Recognition, mm-Wave Radar, Contactless Interface

1 Introduction

Robotic arms have become increasingly important in recent years, especially in fields such as industrial automation [1], healthcare [2], agriculture [3], and service robotics [4]. They support continuous operations, highly efficient, and capable of operating in hazardous environments that may be unsafe for humans. With their growing adoption in daily life, there is a pressing need for more intuitive and efficient ways to interact with them-especially in scenarios where full automation is neither feasible nor desirable, and human judgment remains essential. For example, in assistive care [5], elderly or disabled individuals intuitive and accessible means to control a robotic arm for tasks like feeding, picking up objects, delivering, and lifting. In medical settings, healthcare personnel can rely on teleoperated robots [6] to adjust cameras or lighting equipment and deliver medicine or tools, all without physical contact, ensuring sterile operation. In hazardous environments such as areas with toxic chemicals, high temperatures, or radiation [7], it becomes essential for human operators to remain in safe zones while controlling robotic arms remotely to perform critical tasks.

1

^{*}Both authors contributed equally to this research

Common ways to control robotic arms include physical devices [8] such as buttons, joysticks, or remotes. These methods are mature and reliable, and they provide precise control in many applications. However, they may not be intuitive for general users and often require a certain level of training. Touchscreen interfaces [9] offer a more visual and flexible way to interact with robotic systems, but they still require manual operation and sustained attention, which may not be suitable for users without computer skills. Voice control [10, 11] provides a contactless alternative, yet it can be affected by noise, accents, and privacy concerns. Gesture recognition [12] offers a natural and contactless solution for a wide range of users, providing a seamless human-machine interaction experience. This makes it especially useful in situations where hands may be occupied, hygiene is important, or quick reactions are needed. Moreover, compared to other modalities like images, radar-based gesture recognition brings unique advantages such as enhanced privacy protection, robustness to environmental variations, and reliable operation under poor lighting or occlusion [13].

1.1 Contributions

In summary, this work presents a gesture-controlled robotic arm integrating millimeter-wave (mm-wave) radar and behavior trees. The main contributions of this work are:

- End-to-end integration of gesture recognition and robotic control: Unlike most radar-based gesture recognition research [13–19] that focuses only on classification, we develop a complete system where nine distinct hand gestures recognized by a robust, contactless mm-wave radar pipeline are translated into real-time commands. By employing behavior trees to define robot tasks triggered by gestures, our system allows intuitive human-robot interaction and easy customization of robot behaviors without requiring specialized programming expertise.
- Practical demonstration through real-world case studies: We validate the system's effectiveness and reliability via several tasks, including picking up, moving, and placing a cup, pouring water, and executing emergency stops, fully controlled by hand gestures.

2 Methods

This part involves (1) the proposed architecture of gesture recognition in Section 3.1, (2) hardware architecture in Section 3.2, and (3) overall software architecture in Section 3.3.

2.1 Gesture Recognition – Proposed Architecture

As depicted in Fig 2, the data processing pipeline converts raw radar input into the point cloud representation. Starting from a 3D matrix of size $N_c \times N_{\rm chirp} \times N_{\rm chan}$ (number of channels × number of chirps × number of antenna elements), a 2D-FFT is applied per channel and summed into an RDM. MTI (Moving Target Indication) and CFAR are applied to filter out static clutter and isolate moving gesture targets. An FFT across the $N_{\rm chan}$ dimension yields the angle map (angle-amplitude map), from which Cartesian coordinates are calculated. Each frame outputs a list of detected objects with peak, range (distance), Doppler, x, and y features for further analysis. The

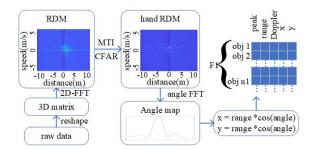


Figure 2: Radar data processing methods.

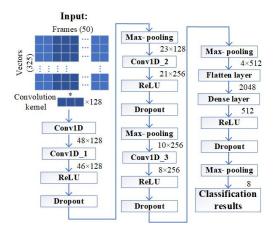


Figure 3: 1D-CNN for gesture recognition.

complete processing chain is implemented on-chip and runs on the radar SoC (System on Chip).

To simplify the algorithm and accelerate the gesture recognition process, a one-dimensional Convolutional Neural Network (CNN) is applied, which is illustrated in Fig 3. The maximum number of frames is 50, and the maximum number of detected objects per frame is 65, resulting in each gesture sample being represented as a 50 × 325 matrix. The input tensor (50, 325) first passes through two consecutive Conv1D layers with 128 filters (kernel size 3), reducing the shape to (46, 128), followed by ReLU, dropout, and max pooling, downsampling the time dimension to 23. A third Conv1D layer with 256 filters (size 3) and a fourth with 512 filters further compress the feature map to (4, 512) through similar post-processing. The output is then flattened to a 2048-dimensional vector, mapped to 512 units via a dense layer with ReLU and dropout, and finally to an 8-dimensional vector representing gesture class logits. The entire process is implemented in real time using buffer management.

2.2 Hardware Architecture

As shown in Fig 1, the hardware setup includes a UR5 robot with a Robotiq 2f-85 gripper mounted on a linear guide for extended motion. The 6-DoF (Degrees of Freedom) UR5, combined with the linear guide, increases the robot's workspace. A laptop controls the entire system, and an AWR1642 radar with 2 TX and 4 RX antennas is used for gesture recognition.

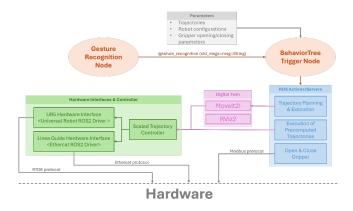


Figure 4: Overall software architecture.

2.3 Overall Software Architecture

The robotic cell in Fig 4 simulates an industrial scenario where a human operator controls tasks using radar-based gestures, typically a pick-and-place operation. Each gesture triggers a behavior tree, representing either a full task or a single action, depending on the control level. The operator completes the task by performing the correct gesture sequence.

The entire control stack is based on the third-party software frameworks and libraries listed in Table 1.

The architecture includes a dedicated ROS 2 node, Gesture Recognition Node, which processes radar data to detect gestures using the algorithm in Section 2.1. It publishes a string identifier to the topic /gesture_recognition. Another node, BehaviorTree Trigger Node, subscribes to this topic and triggers the corresponding behavior tree using the BehaviorTree.CPP library. Each tree can represent a single skill or a sequence for complex tasks. These skills typically involve calling ROS services or actions that plan a trajectory to reach a desired robot configuration or pose, execute precomputed trajectories and open/close the robot gripper. The skill set can be extended to include features like impedance control, force/torque control, or visual servoing for better adaptability. Each service/action is parameterized based on the task (e.g., target pose or gripper state). Trajectories are planned online via MoveIt 2 using a digital twin shown in Fig 1 for collision-free motion. A controller executes the trajectory with fine interpolation and sends real-time commands to the hardware interface. It also supports dynamic speed scaling from 0% (full stop) to 100% (nominal trajectory speed) based on human proximity, ensuring compliance with ISO/TS 15066 [25].

3 Experimental Activities

In this part, the setup is presented, 5 case studies are explored under this architecture, and the results are given.

3.1 Experimental Setup

The initial dataset, adopted from previous experiments, consisted of nine gesture classes, with approximately 2,000 samples per class. These data were collected under highly controlled laboratory conditions, where only the hand was positioned above the radar sensor and background interference was minimized. Although the training

performance appeared very promising, during field testing, significant interference from nearby humans and electronic devices was observed when the radar was deployed in a real-world environment.

Therefore, additional data were collected under realistic conditions and incorporated into the dataset. 5 participants were involved. The composition of the dataset is shown in Table 2. Firstly, 30% of the data collected in the noisy environment is used as the test set. The other data was divided into training and validation sets in a ratio of 7:3.

3.2 Case Studies

Based on the hardware and processing infrastructure previously described, five different tests have been carried out:

- Test 1: The human performs specific gestures to activate each individual robot action required to complete a full pick-and-place task with a glass.
- **Test 2:** Same as test 1, but the robotic arm is behind the human, introducing interference.
- Test 3: The human uses gestures to trigger a sequence of robot actions that pour water from a bottle into a glass.
- Test 4: The human uses the "S" gesture to emergency stop the robotic arm.
- Test 5: The human continuously controls the positioning of the linear guide through hand movements (velocity control).

More specifically, in Tests 1 and 2, the gesture sequence was as follows:

- Performing the "swipe right" gesture moves the robotic arm to the right.
- "Swipe CCW (counterclockwise)" opens the gripper.
- "Down" moves the arm downward.
- "Swipe CW (clockwise)" closes the gripper.
- The arm then moves to the right position again.
- "Swipe left" moves the arm to the left.
- The "S" gesture triggers the object placement operation.
- The arm then moves to the left position again.
- Finally, the "up" gesture commands the robotic arm to return to its home position.

In Test 3, a more complex task involving multiple objects was performed:

- The "swipe right" gesture moves the robotic arm to the right.
- The "down" gesture initiates the grasping of a glass.
- "Swipe left" moves the arm to the left.
- The "Z" gesture places the glass on the table.
- The arm then moves right again.
- The "X" gesture commands the arm to pick up a bottle.
- After that, it moves left once more.
- The "Swipe CW" gesture places the bottle next to the glass.
- The "Swipe CCW" gesture triggers the pouring action to pour water into the glass.
- Finally, the "up" gesture returns the robotic arm to the home position.

In Test 4, the emergency stop is achieved by rapidly scaling the robot's velocity down to zero using the trajectory controller

Library	Description
ROS 2 [20, 21]	Middleware for modular development of robotic applications. Enables communication between software components (nodes) through <i>topics</i> , <i>services</i> , and <i>actions</i> .
ros2_control [22]	ROS 2 framework for real-time robot control. Uses hardware interfaces to read joint states and send commands, connecting physical hardware with control logic.
BehaviorTree.CPP [23]	C++ library for building and executing behavior trees.
MoveIt 2 [24]	ROS 2 platform for motion planning, manipulation, and control.

Table 1: Libraries and frameworks used in proposed architecture.

Table 2: Composition of the dataset

Settings	Samples
only hand in front of the radar	≈ 2000/class ≈ 200/class
robotic arm at the back of the human	$\approx 200/class$

described in Section 2.3. The velocity control logic in Test 5 operates as follows: each recognized hand gesture corresponds to a specific robot velocity command (for example, moving the linear axis left or right at a predefined speed). More specifically, the commanded velocity v(t) is determined by the rule:

$$\begin{cases} v(t) = v_{\text{nom}} \cdot \lambda(t) \\ \lambda(t) = 0.5 \cdot \left(1 + \cos\left(\pi \frac{\min(t, T)}{T}\right)\right) \end{cases} \tag{1}$$

where t is the elapsed time since the gesture was recognized, $v_{\rm nom}$ is the nominal velocity associated with the gesture, $\lambda(t)$ is a scaling factor that smoothly decreases the velocity to zero over time, and T is the duration after which the velocity reaches zero. Each time a gesture is detected, the system selects the corresponding $v_{\rm nom}$ and resets t to zero. As time progresses without new gesture input, the velocity gradually decreases to zero according to Eq. 1, ensuring safety. However, if the gesture is continuously maintained, the velocity remains close to $v_{\rm nom}$.

3.3 Experimental Results

After training the model with the updated data, the test results of the data collected in the field improved significantly, indicating that the gesture recognition system was robust and ready for practical deployment. The validation confusion matrix is shown in Fig 5. And the evaluation metrics of test set are as follows: Accuracy: 0.9368, Recall: 0.8426, F1-score: 0.8544. All case studies were successfully completed, and the interference in Test 2 did not influence at all. A demonstration video was recorded for each scenario.

4 Conclusion and Future Work

Our system uses mmWave radar to recognize hand gestures for touchless robotic arm control. Gestures trigger robot actions via behavior trees, offering robustness and privacy even in noisy environments. While the current system only triggers robot actions through a codified gesture set, future developments will focus on enabling seamless, real-time control, allowing users to intuitively

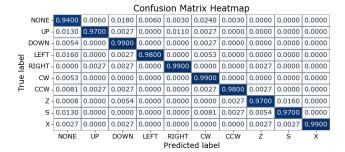


Figure 5: Confusion matrix of gesture recognition.

guide the robot's movements and operating range with enhanced freedom and precision. This could be achieved by expanding the number of recognizable gestures and enhancing the control logic. Given the nature of gesture control, we will also apply it to smart home scenarios such as music playback and lighting control.

Acknowledgments

This work was supported by EU Project HOLDEN, grant agreement 101099491.

References

- S Sanjeev Vyaas, G Shamsudeen Babu, SS Raghulnath, P Manikandan, and E Joel. Robotic arm for industry automation. *Intelligent Robots and Cobots: Industry 5.0 Applications*, pages 203–220, 2025.
- [2] Maria Kyrarini, Fotios Lygerakis, Akilesh Rajavenkatanarayanan, Christos Sevastopoulos, Harish Ram Nambiappan, Kodur Krishna Chaitanya, Ashwin Ramesh Babu, Joanne Mathew, and Fillia Makedon. A survey of robots in healthcare. Technologies, 9(1):8, 2021.
- [3] Tantan Jin and Xiongzhe Han. Robotic arms in precision agriculture: A comprehensive review of the technologies, applications, challenges, and future prospects. Computers and Electronics in Agriculture, 221:108938, 2024.
- [4] Juan Angel Gonzalez-Aguirre, Ricardo Osorio-Oliveros, Karen L Rodríguez-Hernández, Javier Lizárraga-Iturralde, Ruben Morales Menendez, Ricardo A Ramirez-Mendoza, Mauricio Adolfo Ramirez-Moreno, and Jorge de Jesus Lozoya-Santos. Service robots: Trends and technology. Applied Sciences, 11(22):10702, 2021.
- [5] R Sather, Mahsa Soufineyestani, Nabiha Imtiaz, and Arshia Khan. Assistive robots designed for eldery care and caregivers. *International Journal of Robotics* and Control, 3(1):1, 2021.
- [6] David Silvera-Tawil. Robotics in healthcare: A survey. SN Computer Science, 5(1):189, 2024.
- [7] James Trevelyan, William R Hamel, and Sung-Chul Kang. Robotics in hazardous applications. Springer handbook of robotics, pages 1521–1548, 2016.
- [8] Reduanur Rahman, Md Sajid Rahman, and Jillor Rahman Bhuiyan. Joystick controlled industrial robotic system with robotic arm. In 2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON), pages 31–34. IEEE, 2019.

4

- [9] Y Rabhi, M Mrabet, F Fnaiech, P Gorce, I Miri, and C Dziri. Intelligent touchscreen joystick for controlling electric wheelchair. *Irbm*, 39(3):180–193, 2018.
- [10] Rashed S Kanash, Seyed E Alavi, and Ali A Abed. Design and implementation of voice controlled robotic arm. In 2021 International Conference on Communication & Information Technology (ICICT), pages 284–289. IEEE, 2021.
- [11] Riccardo Adamini, Nicholas Antonini, Alberto Borboni, Simone Medici, Cristina Nuzzi, Roberto Pagani, Alberto Pezzaioli, and Cesare Tonola. User-friendly human-robot interaction based on voice commands and visual systems. In 2021 24th International Conference on Mechatronics Technology (ICMT), pages 1–5, 2021.
- [12] Szilard Bularka, Roland Szabo, Marius Otesteanu, and Mircea Babaita. Robotic arm control with hand movement gestures. In 2018 41st International Conference on Telecommunications and Signal Processing (TSP), pages 1–5. IEEE, 2018.
- [13] Shahzad Ahmed, Karam Dad Kallu, Sarfaraz Ahmed, and Sung Ho Cho. Hand gestures recognition using radar sensors for human-computer-interaction: A review. Remote Sensing, 13(3):527, 2021.
- [14] Jih-Tsun Yu, Yen-Hsiang Tseng, and Po-Hsuan Tseng. A mmwave mimo radar-based gesture recognition using fusion of range, velocity, and angular information. IEEE Sensors Journal, 24(6):9124–9134, 2024.
- [15] Haoming Liu and Zhenyu Liu. A multimodal dynamic hand gesture recognition based on radar-vision fusion. *IEEE Transactions on Instrumentation and Measurement*, 72:1–15, 2023.
- [16] Guiyuan Zhang, Shengchang Lan, Kang Zhang, and Linting Ye. Temporal-range-doppler features interpretation and recognition of hand gestures using mmw fmcw radar sensors. In 2020 14th European conference on antennas and propagation

- (EuCAP), pages 1-4. IEEE, 2020.
- [17] Geethika Bhavanasi, Lorin Werthen-Brabants, Tom Dhaene, and Ivo Couckuyt. Patient activity recognition using radar sensors and machine learning. *Neural Computing and Applications*, 34(18):16033–16048, 2022.
- [18] Song Wang, Zecheng Li, Rongjie Huang, Ronghao Wang, Junjie Li, and Zhiwei Xu. Hand gesture recognition scheme based on millimeter-wave radar with convolutional neural network. In *IET Conference Proceedings CP779*, volume 2020, pages 488–492. IET, 2020.
- [19] Dariush Salami, Ramin Hasibi, Sameera Palipana, Petar Popovski, Tom Michoel, and Stephan Sigg. Tesla-rapture: A lightweight gesture recognition system from mmwave radar sparse point clouds. *IEEE Transactions on Mobile Computing*, 22(8):4946–4960, 2022.
- [20] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. Science robotics, 7(66):eabm6074, 2022.
- [21] Steven Macenski, Alberto Soragna, Michael Carroll, and Zhenpeng Ge. Impact of ros 2 node composition in robotic systems. IEEE Robotics and Autonomous Letters (RA-L), 2023.
- [22] OSRF. ros2_control documentation. https://control.ros.org/rolling/index.html, 2024. Accessed: May 2025.
- [23] Davide Faconti. BehaviorTree.CPP, 2024. Accessed: May 2025.
- [24] Sucan Ioan A. and Chitta Sachin. MoveIt, 2024. Accessed: May 2025.
- [25] ISO Central Secretary. ISO/TS 15066:2016 Robots and robotic devices Collaborative robots. Standard, International Organization for Standardization, Geneva, CH, February 2016.