IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 July 2025, accepted 30 July 2025, date of publication 5 August 2025, date of current version 15 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3595995

RESEARCH ARTICLE

Human-Aware Robot Collaborative Task Planning Using Artificial Potential Field and DQN Reinforcement Learning

JAYESH PRAKASH[1], SAM ALTNJI[2], KARTHICK THIYAGARAJAN[3], (Senior Member, IEEE), JOGESH S. NANDA[1], ABHIJIT BISWAS[1], AND ABHRA ROY CHOWDHURY[1], (Senior Member, IEEE)

[1]Department of Design and Manufacturing, Indian Institute of Science (IISc), Bengaluru, Karnataka 560012, India
[2]Department of Design, Production & Management, Faculty of Engineering Technology, University of Twente, 7522 Enschede, The Netherlands
[3]Smart Sensing and Robotics Laboratory (SensR Lab), Centre for Advanced Manufacturing Technology, Western Sydney University, Kingswood, NSW 2747, Australia

Corresponding author: Abhra Roy Chowdhury (abhra@iisc.ac.in)

This work is supported by the CORE grant under the Department of Science and Technology (DST), Government of India, under Grant No. CRG2021006698.

ABSTRACT This paper presents a novel way for Robot-Robot-Human interaction in a shared workspace for collaborative tasks and uses a multi-modal means of communication that includes hand gestures, voice commands, end-effector gestures, and marker tracking. The system consists of a human operator working along with a Task robot (UR5) and a helper robot (OpenManipulatorX) to perform assembly and disassembly tasks. A Deep Q Network (DQN) reinforcement learning model is used to train the robot to perform the goal reaching task while avoiding obstacles to ensure safety. The DQN algorithm makes use of the end-effector position and the relative positions with the goal and obstacles to train a policy that guides the robot arm safely. Then 4 different training models are created and their ability to avoid obstacles and reach the goal are compared along with the point-to-point Bezier interpolation path planning method in different scenarios such as varying height, size, and number of obstacles. The proposed system has been simulated and then experimentally validated. Experimental results show that DQN trained model performed better than Bezier interpolation in reaching the final goal position with an accuracy of 74mm while avoiding obstacles at the same time in a shared environment. It is also observed that of the different trained models, the model with a larger action space and reduced observation space gave better results compared to others in terms of accuracy and goal completion rate. Also, from experimental data its observed that Improved Artificial Potential Field (IAPF) only took 4.7s as the median time to reach the goal whereas Goal Directed Approach (GDA) took 7.62s and Rapidly Exploring Random Tree Star (RRT*) took 6.22s in different scenarios.

INDEX TERMS Cobot, multi-robot, human-robot-interaction, robot-robot-interaction, gesture recognition, audio commands, DQN, reinforcement learning.

I. INTRODUCTION

Collaboration between humans and robots is emerging as a field of interest for various industries and researchers. This paper proposes to create a mode of intuitive and user-friendly collaboration between human operators and multiple robots in a shared workspace. We propose to use a multi-modal mean of communication between the Human and robots that includes gesture commands, voice commands, pose tracking, and marker tracking. Previous works in this field either make use of only gestures [1], [2] or only voice commands [3], [4]. Some works include both audio and visual perception, but the visual perception tracks the human lips motion to assist in recognition of voice commands [5], rather than using gestures to specify a task. Other existing work makes use of sound localization methods, along with visual feedback

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Huei Cheng.

for tracking or navigation [5], [6]. Some works included audio and visual cues as feedback for the operator rather than for the control of the robot [7]. Other research used Hierarchical reinforcement learning to move the robot to goal by breaking the path into stages [8]. Proximal Policy Optimization(PPO), Deep Deterministic Policy Gradient(DDPG), Hindsight Experience Replay(HER), Actor-critic and other Reinforcement learning models are also explored for control of robot arm [9], [10], [11]. Reward optimization [12], [13] and shaping approaches [14] are used to assist in reaching the goal. Potential rewards functions have been explored [15]. Most of these works focus on goal completion rather than safety. To ensure safety, velocity scaling approaches [16], potential fields [17], [18] time series prediction of human motion [19] and creating layers of safety zones with different safe velocities around robot [20] are made use of. Some works deal with obstacle avoidance using Deep Q Network(DQN) [21] but they did not consider involvement of human agent and multi-robots.

This proposed collaboration between robots is inspired by the collaborative interaction between Coral Grouper and the Moray eel in the sea [22], [23]. Coral groupers usually hunt alone, but sometimes their prey might escape and hide in cracks of the reef. So, Coral groupers collaborate with Moray eel. Grouper will point its nose toward the prey and shake its body side by side, to notify the eel about the prey. Now the eel will go inside the crack and get the prey. This increases the chances for both to get the food. Similar problems are faced when dealing with industrial robots in assembly task [16]. The robots are designed for heavy tasks and are huge in size. The large end-effector is unable to reach a lot of small gaps or under the object for various tasks. Also, sometimes just one arm is not enough to do even the simplest tasks of pick and place. To help with these problems, a small helper robot is introduced (OpenManipulatorX) that is able to reach small gaps and helps the big robots by holding the job while the big robot disassembles the components from it. In addition to this, the helper robot also helps the human operator working in the shared workspace by fetching its tools through gesture and voice commands. For the task, the standard assembly/disassembly task board is used being provided by NIST for IROS Robot Grasping and Manipulation Competition. The task robot is trained using the DQN algorithm reinforcement learning in a virtual environment to reach the goal position while avoiding obstacles.

The main contribution of this paper is as follows.

- A nature inspired collaboration framework between two robots-and one human worker
- Multi-modal (audio, gesture pose tracking and marker tracking) means of communication for assisting human workers.
- A DQN reinforcement learning model for a 6-axis robot arm for goal reaching task and obstacle avoidance for a safe local dynamic path planning.

The paper is structured as follows. Section I gives an introduction to the proposed framework, and Section II explains the purposed framework and methodology in which the proposed framework is developed. Section III explains path planning and DQN reinforcement learning. Section IV explains the simulation and the experimental results.

II. PROPOSED METHODOLOGY

The proposed framework shown in Fig.1 consists of decentralized blocks for a 'Task' robot and a 'Helper' robot working in a shared workspace along with a human operator (worker). The input node is the human operator for both. The user commands them through gesture and voice commands. A DQN reinforcement learning model will read the robot and environment states and pass on the actions to the Task robot. The camera mounted on Task robot's end effector will track the human worker hand and the helper robot in his field of view and provide the observations to the DQN network which will then plan a safe path to avoid any collisions. The helper robot will track the marker on the end effector of the Task robot and assist the task robot in tasks requiring two arms. The proposed framework consists of the following systems.

A. ROBOT AGENTS: TASK ROBOT (R1), HELPER ROBOT (R2)

A 6-DoF robot arm (UR5) mounted with an RGB-D camera (Intel RealSense) is being used as a Task robot. The task is to disassemble the components from a task board designed by NIST for IROS RGMC and put all the disassembled components in a container. A camera is also mounted to the end effector which uses IR stereo vision to find the depth of all points in the workspace and then the coordinates of detected objects are transformed to world frame coordinates using forward kinematics frame transforms on the robot link. A 4-DoF OpenManipulatorX (500g payload capacity) equipped with a RGB webcam mounted above the end effector is used as a helper robot. The objective of this helper robot is to assist the human operator and the Task robot in their tasks.

B. INDUSTRIAL SPACE FORCE MODELLING

The motion of the human worker is modeled based on the extended social force model (ESFM) [24]. This allows us to predict the desired direction the human will move in the future time step and accordingly plan guidance framework. The resultant force acting on the human can be formulated as

$$F^r = \alpha f_{r,d}^{goal}(D_n^{goal}) + \beta f_{r,p}^{goal}(D_n^f) + (\gamma f_r^{ped} + \delta f_r^{obs}) \quad (1)$$

where α, γ and β parameters are the weights of the respective forces, and F_g is the resultant force of the group, $f_{g,d}^{goal}(D_n^{goal})$ is the attractive force until the final destination of the group, f_{gped} and f_{gobs} are the repulsive forces from co-workers and obstacles respect to the group. To estimate the goal D_n, the Bayesian approach is used to estimate the probability of a desired goal g_t^i, of the pedestrian i based on a past observation history, X_{obs}^i spanning $t = 1$ to $t = T_{obs}$.

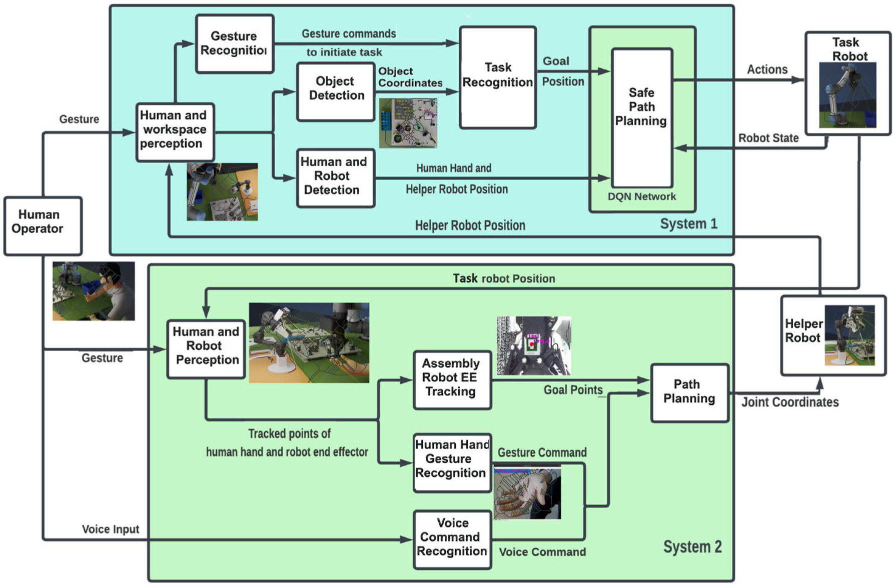


FIGURE 1. Framework for robot-robot-human collaboration in overlapping workspace with multimodal audio-visual communications.

$P(g_t^i|X_{obs}^i)$ is the posterior probability of each goal, given an observation history X_{obs}. The above-mentioned social force model is used to estimate the position of the human in a future time $t+k$. Let the state of the worker at $t+k$, $s_{(t+k)} = (x_{(t+k)}, y_{(t+k)}, \theta_{(t+k)})$. Based on the proxemics and preferred formation, the online goal for the robot can be estimated. This is the most basic case. Scenarios where there are static or dynamic obstacles are discussed in the next section.

C. SAFETY METRICS

Human Safety and Comfort Indices (HSCI) [16], [25], [26] modified to fit our scenario. An Industrial Interaction Index (3I) is used to measure the psychological safety of humans. The 3I value should be less than the psychological threshold T_c

$$3I = \max_{i=1:N} exp\left(-\left(\left(\frac{x_r - x_i^p}{\sqrt{2}\sigma_0^{px}}\right)^2 + \left(\frac{y_r - y_i^p}{\sqrt{2}\sigma_0^{py}}\right)^2\right)\right) \tag{2}$$

The $\sigma_{px} = \sigma_{py} = 0.5dc$ value is selected according to the Hall's personal space criterion. dc is 0.9 meters. The modified Direction Index (DI) [24] which measures the industrially compliant orientation of robot, also includes the following user.

$$DI = \min_{i=1:N} \frac{2 + \cos(\varphi_i) + \cos(\beta_i)}{4 * \left(\sqrt{\left(x_i^p - x_r\right)^2 + \left(y_i^2 - y_r\right)^2}\right)} + \frac{\sin(\theta_{ru})}{\sqrt{(x_u - x_r)^2 + (\psi_u - y_r)^2}} \tag{3}$$

where θ_{ru} is the angle between the worker and the robot. Robot must occupy the r space as follows [27], [28].

- Intimate distance: Distances between people inside the interval [0-45 cm]
- Personal distance: Distances between people inside the interval [45 cm-1.22 m]
- Social distance: Distances between people inside the interval [1.22-3 m]
- Public distance: Distances between people are 3m away (> 3 m)

$$S = \left\{x \in R^2 \setminus (\mathcal{O} \cup \mathcal{C}) | 0.75 < d(x, p_c) < 3\right\} \tag{4}$$

$$O = \left\{x \in R^2 \setminus C | d(x, r_0) < 1\right\} \tag{5}$$

$$C = \left\{x \in R^2 | d(x, p_i) < 0.75\right\} \tag{6}$$

Assume the robot TCP [29] has a radius of r_d and can be represented as a circle of $2r_d$ of diameter, with center on the robot's position $R = \{x \in R^2 | d(x, r) < 0.5\}$, whose area is $|R|$ = The area metric for the companion task can be computed for each human's position pc and robot's position p_r, and is formulated as follows

$$P(p_r, p_c) = \frac{1}{|\mathcal{R}|}\int_{E \cap R} dx + \frac{1}{2|\mathcal{R}|}\int_{A \cap R} dx \in [0, 1] \tag{7}$$

IAPF combines multiple obstacle repulsive forces, allowing adjustable influence distances, distinguishing it from traditional methods, which reduces local minima issues [30], [31]. To improve path quality, optimization techniques are used, reducing abrupt changes, and achieving a smoother, energy-efficient, and reliable robot navigation path. The `human operator (H)`, `Task Robot (R1)`, and `Helper robot (R2)` will be working together in an intersecting workspace denoted as S_{shared}. The reachable workspace of human worker is dynamic, since he is having unpredictable behavior, movements, and varying physical properties for different operators. The position of human hand, helper robot end effector and end goal position are given by ${}^{W}P_{hum}$, ${}^{W}P_{H}$ and ${}^{W}P_{goal}$ respectively. When any of ${}^{W}P_{hum}$, ${}^{W}P_{H}$ and ${}^{W}P_{goal}$ is within the shared space S_{shared}, the path of the robot must be checked for any collision and adjusted to avoid them. These coordinates are used as observation states in the reinforcement learning environment, where the agent i_e. Task robot moves in the environment and reads these observation states. A model trained with the DQN algorithm makes use of this observation to decide the respective actions to allow the robot to reach the goal while avoiding the obstacles.

III. DQN-REINFORCEMENT LEARNING

DQN Reinforcement learning [32] approach is used to train the robot to reach the desired goal points while avoiding the obstacles in the shared workspace, that are the helper robot and the human operator's hand. Deep Q Network was used to train this model. A custom OpenAI gym environment is created that communicates with the gazebo environment and reads the environment observations and applies actions to it. The state s_t is defined as an array of 12 elements, where the first 3 elements define the end-effector position (${}^{W}P_{ee}$), 3 elements define the end-eff position relative to the goal (${}^{W}P_{ee}$ - ${}^{W}P_{goal}$), 3 elements define the end-eff position relative to helper robot end-effector (${}^{W}P_{ee}$ - ${}^{W}P_{H}$) and 3 elements define the end-eff position relative to human operator's hand (${}^{W}P_{ee}$ - ${}^{W}P_{hum}$). This state space is discretized into small bins in each axis. The state is defined as

$$s_t = ({}^{W}P_{ee(1*3)}, {}^{Goal}P_{ee(1*3)}, {}^{H}P_{ee(1*3)}, {}^{hum}P_{ee(1*3)}) \tag{8}$$

The action space is defined as a set of 27 actions. The actions increment and decrement the Task robot's end effector in cartesian coordinates x,y and z. The 27 actions represents the 27 combinations of x,y,z having values -1,0 and 1, which represents, decrement by 1 cm, stay still and increment by 1 cm respectively. The geometric IK solver is used to get the joint coordinates for these end effector states. The Q-function estimates the long term rewards for particular actions, at any state. The optimal policy π is decided based on the Q-values that maximizes the long-term reward. Then at a given state, the policy decides the appropriate actions based on the Q-values. The Q-function is updated using the following equation:

$$Q(s, a) \longleftarrow Q(s, a) + \alpha[r + \gamma \max(Q(s', a')) - Q(s, a)] \tag{9}$$

where $Q(s, a)$ is the current estimate of the Q-function for state s and action a. r is the reward received for transitioning to state s'. s' is the new state reached after taking action a in state s. a' is the action taken in the next state s'. $\max(Q(s', a'))$ is the maximum expected reward for all actions a' in the next state s'. γ is the discount factor, which determines the importance of future rewards. A discount factor of 0 will make the agent only consider immediate rewards, while a discount factor of 1 will make the agent consider all future rewards equally. α is the learning rate, which determines how much the agent adjusts the Q-function in response to new information.

The reward r is received for performing action a and reaching state s' from state s. The reward function is formulated as follows:

$$r = (r_{goal} + r_{col} + r_{bound} + r_{step}) \tag{10}$$

where r_{goal} is the reward received for reaching the goal, r_{col} is the negative reward for collision, r_{bound} and r_{step} are negative rewards for reaching the boundary constraints in joint and task space and negative reward for taking each number of steps respectively. r_{goal} consists of a positive reward for reaching the goal and potential reward for getting closer to the goal in each step compared to the previous closest distance reached. These rewards were described as:

$$r_{goal} = r_g + r_{\phi} \tag{11}$$

$$r_g = \begin{cases} R_s*(G_{th} - d_{goal}) & \text{if} d_{goal} < G_{th} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

r_g is the goal reaching reward and R_s is the reward scale factor and G_{th} is the distance threshold from goal to consider it as goal reached.

$$r_{\phi} = r_{distance} + r_{advancement} \tag{13}$$

r_{ϕ} is the potential based reward that directs the agent to move towards the goal.

$$r_{distance} = \begin{cases} R_s*(d'_{goal} - d_{goal}) & \text{if} d_{goal} < d'_{goal} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$r_{advancement} = (x'_{goal} - x_{goal}) + (y'_{goal} - y_{goal}) + (z'_{goal} - z_{goal}) \tag{15}$$

This $r_{advancement}$ will produce a negative reward when the agent moves far from the target and ensure the motion is directed towards the goal. d_{goal} is the distance between robot end effector and goal point, d'_{goal} is the closest distance reached so far. The negative reward for collisions is given by:

$$r_{col} = r_H + r_R + r_T \tag{16}$$

where r_H, r_R and r_T are negative rewards for hitting the human operator's hand, helper robot's end-effector and base

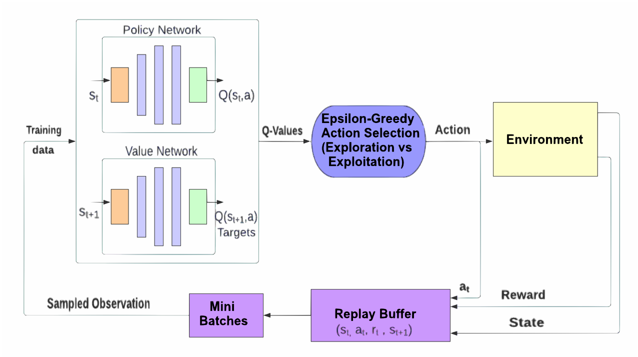


FIGURE 2. **DQN model.**

table respectively. These negative rewards are given by:

$$r_H = \begin{cases} -R_p & \text{if} d_H < O_{th} \\ -(O_w - d_H) & \text{if} O_{th} < d_H < O_w \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

$$r_R = \begin{cases} -R_p & \text{if} d_R < O_{th} \\ -(O_w - d_R) & \text{if} O_{th} < d_R < O_w \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

$$r_T = \begin{cases} -R_p & \text{if} z_{ee} < z_{th} \\ -(z_w - z_{ee}) & \text{if} z_{th} < z_{ee} < z_w; d_{goal} > G_{in} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where, d_H is the distance from human hand, d_R is the distance from the helper robot end effector, and all dimensions are in mm. O_{th} is the safe distance threshold for obstacles, and O_w is the warning zone around the obstacles.z_{th} is the safe distance threshold above the table, and z_w is the warning zone above the table. G_{in} is the radius for sphere of influence around the goal point.R_p is the penalty score awarded for failure. r_{bound} consists of negative rewards for when the joins angles exceed the allowable range, and a negative reward for when the end-effector tries to move to the unreachable cylindrical volume at the center of the Cobot. It is given by:

$$r_{\text{bound}} = \begin{cases} -R_p & \text{if } \sqrt{x_{ee}^2 + y_{ee}^2} < d_r \\ -R_p & \text{if } \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\} \nsubseteq \Theta_{\text{allowable}} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

where d_r is the unreachable inner radius around the robot base. And $r_{step} = -1$ is the negative reward for each step taken to minimize the number of steps taken to reach the goal.

In DQN, neural networks are used to predict the Q-values for each observation state. Two separate networks are built, both having same structure. One of the network called the 'Policy network' or 'Q-Network' predicts the Q-values for all action taken at state 's' i.e $Q(s_t, a)$. The weights of this network are updated at every step. Another network called 'Value Network' or 'Target Network is used to predict the target values $\hat{Q}(s, a)$ for the policy network. The weights of this network are updated after every C steps to provide a static target for the policy network to chase. Fig. 2 shows the structure for a DQN model. Where

- s_t, a_t and r_t are the state, action and reward at time t.
- s_{t+1} is the state reached after taking action a_t in state s_t.
- γ is the discount factor that determines the importance of future rewards.
- ϵ is the probability of selecting a random action (exploration).
- $\hat{Q}(s, a; \theta_{t-1})$ is the action-value function estimate of the target network, which has weights θ_{t-1}.
- D is the replay memory, which stores a fixed-size buffer of transitions (s_i, a_i, r_i, s_{i+1}).
- M is the size of the mini-batch, which is a randomly sampled subset of the transitions in the replay memory.
- y_i is the target value for the i^{th} transition in the mini-batch.
- $\mathcal{L}(\theta_t)$ is the loss function, which measures the difference between the target values and the predicted values of the Q-function.
- C is the number of steps between updates to the target network.

First, we define the action-value function $Q(s, a; \theta)$, where s is the current state, a is the current action, and θ are the weights of the value network. The goal of the DQN algorithm is to learn the optimal action-value function $Q^*(s, a)$ that maximizes the expected cumulative reward. At each time step t, the DQN algorithm updates the weights θ_t of the policy network based on the loss function $L(\theta_t)$, which measures the difference between the predicted action-value $Q(s_t, a_t; \theta_t)$ and the target value y_t. The target value is calculated using the Bellman equation.

$$y_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta_{t-1}) \tag{21}$$

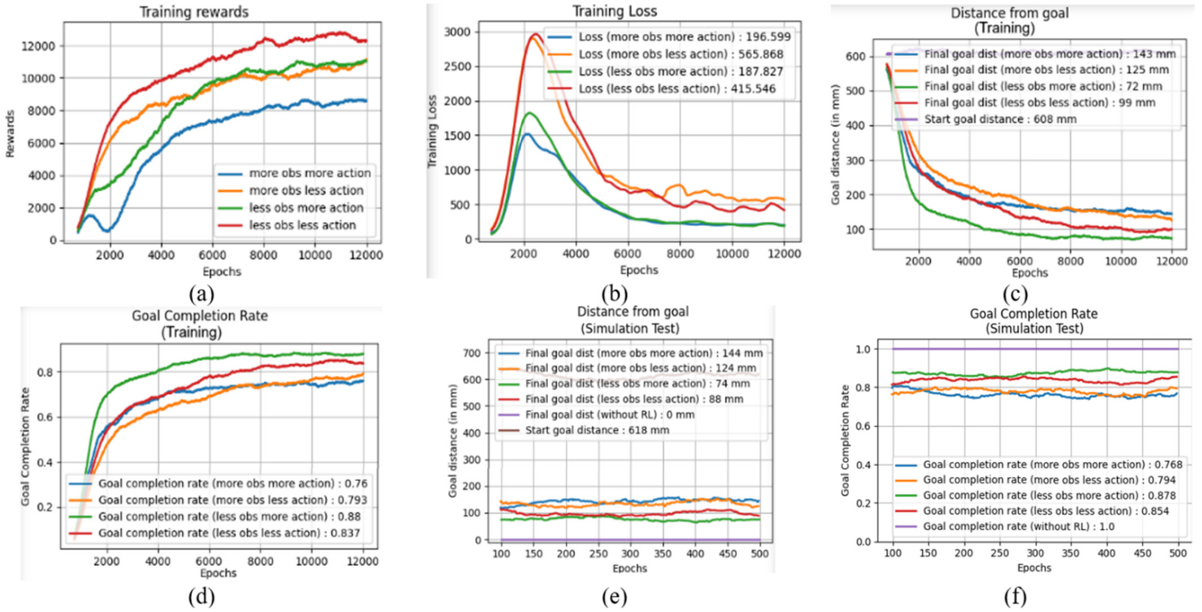
where θ_{t-1} are the weights of the target network, which are periodically updated to be equal to the weights of the policy network. The loss function for the policy network is defined as the mean squared error (MSE) between the predicted action-value and the target value.

$$L(\theta_t) = \mathbb{E}_{s_t, a_t}\left[(y_t - Q(s_t, a_t; \theta_t))^2\right] \tag{22}$$

The Adam optimizer is used to update the weights θ_t based on the gradient of the loss function with respect to the weights:

$$\theta_{t+1} = \theta_t + \alpha \frac{\partial L(\theta_t)}{\partial \theta_t} \tag{23}$$

where α is the learning rate. After getting the Q-functions through the neural network, an ϵ-greedy action selection strategy is utilized to decide between 'Exploration' and 'Exploitation'. This is done to explore the environment, since the immediate best action might not provide the best result in the long run. A decay function is applied to the ϵ so that the weightage of exploration depletes as the network learns better strategies, and starts focusing on exploitation of the strategies already found to provide better results.



(a) (b) (c) (d) (e) (f)

FIGURE 3. (a-d) Comparison of training result in 4 Environments, (e) Simulation result of 4 environment for distance to goal (f) simulation result for goal completion rate of 4 environment.

IV. SIMULATIONS AND EXPERIMENTS

A. EXPERIMENTAL SETUP

The experimental validations were performed using the UR 5 Robot. It runs on Snapdragon 410 and Snapdragon 820 mobile processors, Windows IoT Core, and Android 8. Jetson Nano was used for parallel processing. Intel RealSense camera D435i is mounted on the 3 axis end-effector. We have used Intel® RealSense^{TM} D435 with a frame rate of 30 fps and a resolution of 1280 × 720. Three far-field microphones using Qualcomm® FluenceTM PRO is used for sound localization and recording. Python and MATLAB were used to simulate and validate the individual components of the system. Four different training environments are created to train the robot to reach the goal points while avoiding obstacles, and a comparison is made between the results of each environment. The reward functions and training hyper-parameters remains the same for each of the models. Since the states observed in the environments are continuous space rather than discrete, the observed states are divided into discretized bins. The end effector position w.r.t. world is discretized in 160 evenly distributed values ranging between −800mm to +800mm. On the other hand, the positions w.r.t. goal and obstacles are discretized in 160 unevenly distributed bins ranging from −800mm to +800mm, where the size of each bin starts with 20mm at each end and reduces to 5mm as it approaches the 0 position. The first environment ‘E−1’ uses a large observation space and a large action space. The observation space of first environment is defined in (24) and consists of 27 parameters.

$$S_{E-1} = ({}^{W}P_{ee(1*3)}, {}^{Goal}P_{ee(1*3)}, {}^{H}P_{ee(1*3)}, {}^{hum}P_{ee(1*3)}) \quad (24)$$

whereas the action space ‘A_(E−1)’ consists of 27 actions. In the second environment ‘E−2’ uses a larger observation space but small action space. The observation space of second environment, ‘S_(E−2)’ is same as ’S_(E−1)’. But the action space ‘A_(E−2)’ is constrained to only 7 actions, i.e., forward and backward translation in only one axis at a time and one action of staying still. Thus, reducing the possible choices and making the decision easier. The third environment ‘E-3’ consists of larger action space, but reduced observation space. The goal is to get the maximum useful information using minimum data. Since the end effector can only collide with one obstacle at an instant of time, it was not necessary to always keep track of all the obstacles. Thus, only the distance from the obstacle closest to the end-effector at time step ‘t’ was considered in the observation space. The new observation space is defined as

$$S_{E-3} = ({}^{W}P_{ee(1*3)}, {}^{Goal}P_{ee(1*3)}, {}^{closestobstacle}P_{ee(1*3)}) \quad (25)$$

The fourth environment ‘E−4’ uses the smaller observation space as well as smaller action space, thus making the computation and decision making easier. The observation space ‘S_(E−4)’ is same as ‘S_(E−3)’ and the action space ‘A_(E−4)’ is same as ‘A_(E−2)’

B. TRAINING PARAMETERS

The following parameters were used for the training model:

1) Policy network and Value networks with input size of [1,12] for larger observation space and [1,9] for smaller observation space
2) One fully connected Dense layer with 128 nodes and ReLU activation,

3) One fully connected Dense layers with 64 nodes and ReLU activation
4) An output layer of size 27 with linear activation for larger action space and size 9 for smaller action space
5) Epochs = 12000
6) epsilon = 1.2 with a decay rate of 0.99885 in each epoch and and increment factor of 1.004 at each step in an epoch
7) Batch_size = 256 for sampling from 'Replay Buffer'
8) 'Replay Buffer' with 40000 entries
9) Network_learning_rate = 0.0003
10) Gamma = 0.96 (Discount factor)
11) Weights update rate for value network = 25 steps
12) Reward parameters: $G_{th} = 10$, $R_s = 50$, $R_p = 1000$, $d_r = 150$, $O_{th} = 80$, $O_w = 200.G_{in} = 5$, $z_w = 80$, $G_{in} = 100$.

All the 4 environments were trained for 12000 episodes.

C. EXPERIMENTAL RESULTS

Fig. 3(a-d) shows the comparison of training results and Fig. 3(e-f) shows the comparison of simulation results of the 4 E1- E4 training environments for rewards attained, training losses in the policy network, the final distance reached from the goal point, and the percent of goal completion respectively while learning from 0 to 12000 episodes. It is observed that the environment E3 with smaller observation and large action space have better goal completion rate of 0.878 compared to others (E1 having 0.768, E2 having 0.794 and E4 having 0.854). The trained models were tested for 500 episodes each and compared together along with the results of path planning with Bezier interpolation [33], [34] without safety planning using reinforcement learning. Since the goal of these models was to ensure safety by avoiding collisions, Fig. 3(e) shows the closest distance with the obstacles in the four environments, while a green line represents a safe distance of 80mm from the obstacles. Fig. 3(f) shows the comparison of 4 environments along with Bezier curve interpolation path planning for the final distance reached from the goal and the completion rate. The E3 environment produced the best results with an average final distance of 74mm from the goal while the others E1 having 143mm, E2 having 125mm and E4 having 99mm of distance from the goal. A comparative study explores three algorithms: Gradient Descen t [30], Improved Artificial Potential Field (IAPF) [18], [35], and Rapidly Exploring Random Tree Star (RRT*) [31]. The points, $q_{start}(10, 70, 15)$, $q_{goal}(70, 30, 5)$, and a complex environment with four obstacles: $O_1(30, 55, 20)$, $O_2(25, 30, 20)$, $O_3(52, 50, 20)$, $O_4(55, 20, 20)$. Diverse simulations performed offer comprehensive algorithmic insight.

1) VARYING NUMBER OF OBSTACLES

In Case 1 (Table 4) as shown in Fig. 4(a-d) with varying number of obstacles from 1 to 4, IAPF ($K_{att} = 0.03$, $K_{rep} = 0.1$), performs better than Goal Directed Approach (GDA) and RRT*, reaching the goal in 4.7s. IAPF outperformed GDA by 37% and RRT* by 9% in obstacle-rich environments.

TABLE 1. Varying number of obstacles.

Obstacle Count	Obstacle Positions	Time to Goal-IAPF	Time to Goal-GDA	Time to Goal-RRT*
1	O_1	4.7s	7.56s	5.12s
2	O_{1-2}	4.7s	7.57s	5.81s
3	O_{1-3}	4.7s	7.51s	5.72s
4	O_{1-4}	4.7s	7.62s	6.2s

TABLE 2. Varying size of multiple obstacles.

Size of Obstacles	Time to Goal-IAPF	Time to Goal-GDA	Time to Goal-RRT*
2cm	4.68s	7.65s	5.88s
2.5cm	4.69s	7.61s	6.08s
3.0 cm	4.7s	7.70s	6.20s
3.5 cm	4.72s	7.74s	6.15s

TABLE 3. Multivariate scenario.

Obstacle Size	Time to Goal-IAPF	Time to Goal-GDA	Time to Goal-RRT*
2,2.5,3,3.5cm	4.69s	7.54s	6.14s
2.5,2,3,3.5cm	4.68s	7.58s	6.29s
3,3.5,2,2.5cm	4.72s	7.57	5.77s
3.5,3,2.5,2cm	4.70s	7.74s	5.93s

TABLE 4. Varying height.

Obstacle Size	Time to Goal-IAPF	Time to Goal-GDA	Time to Goal-RRT*
25,20,18,15cm	4.7s	7.61	5.57s
20,25,18,15cm	4.7s	7.61	5.34s
18,15,20,25cm	4.7s	7.61	6.58s
15,18,25,20cm	4.7s	7.61	7.32s

2) VARYING SIZE OF MULTIPLE OBSTACLES

In Case 2 (Table 2) as shown in Fig. 4(e-h), we assessed how varying obstacle radii (2-3.5cm) affected algorithmic performance with four obstacles. IAPF consistently completed tasks in 4.68-4.72s,showcasing its real-world versatility with a 35% and 10% lead over GDA and RRT*, respectively.

3) MULTIVARIATE SCENARIO

In Case 3 (Table 3) as shown in 4(i-m), IAPF excelled, achieving times between 4.68s and 4.72s ($K_{att} = 0.03$, $K_{rep} = 0.1$). GDA required 7.54-7.74s with specific parameters ($K_{rep} = 900$, $K_{att} = 1/150$, $d_0 = 4$), while RRT* performed competitively, ranging from 6.14s to 6.29s. On average, IAPF outperformed GDA by 39% and RRT* by 4%.

4) VARYING HEIGHTS

In Case 4 (Table 4), IAPF consistently reached the goal in 4.7s, unaffected by varying obstacle heights. GDA took 7.61s, and RRT* ranged from 5.34s to 7.32s. On average, IAPF outperformed GDA by 37% and RRT* by approximately 21%, showcasing its adaptability to differing obstacle heights in complex settings.

5) VARYING K (ATTRACTIVE) & VARYING K (REPULSIVE)

In Case 5 (Table 5 & Table 6), comparing IAPF and GDA with varying K_{att} and K_{rep} constants at given locations as shown in Fig. 4(n-p), IAPF consistently reached the goal faster, highlighting its responsive dynamics. GDA had longer

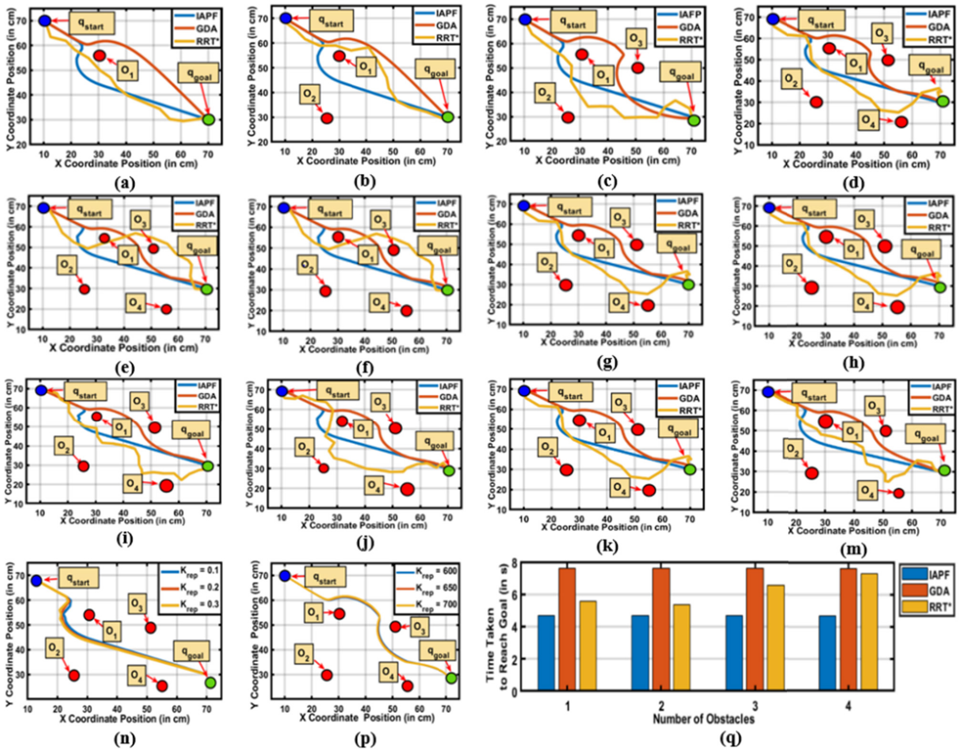

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (m) (n) (p) (q)

FIGURE 4. (a-d) Path planned on varying number of obstacles, (e-h) Path planned on varying sizes of obstacles, (i - m) Paths planned on multiple variations of obstacle sizes, (n) Varying Krep for IAPF (p)Varying Krep for GDA (q) Time to goal for different algorithm.

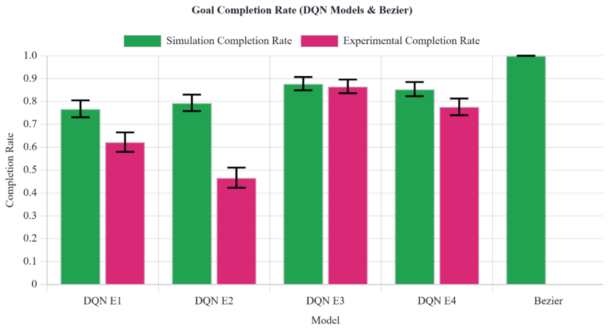


FIGURE 5. Goal completion rate comparison of DQN and Bezier models.

TABLE 5. Varying K(attractive).

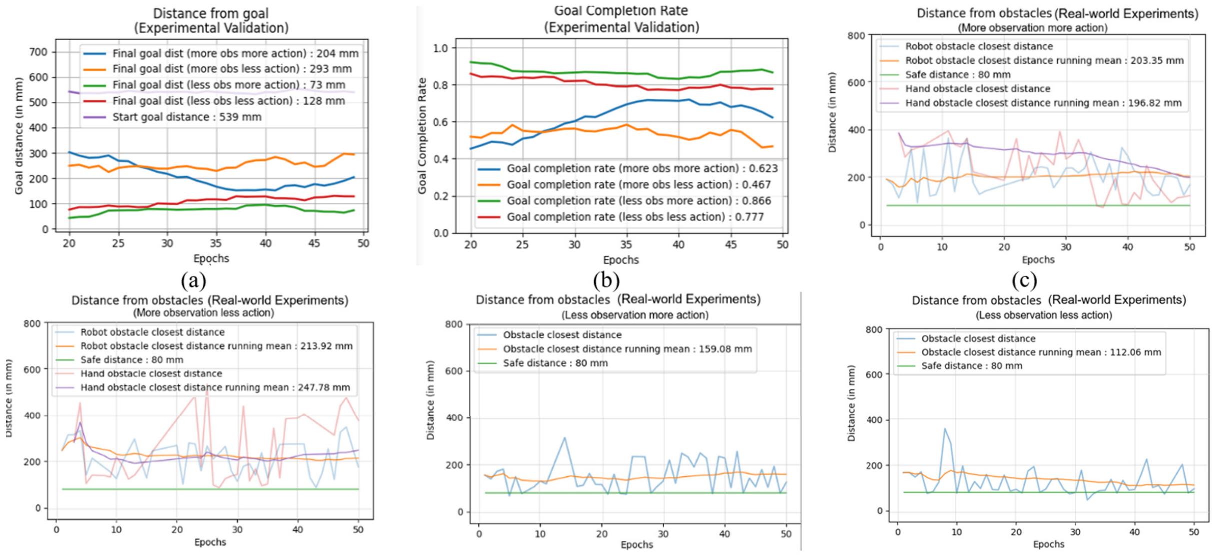
K_{att} IAPF	K_{rep} IAPF	Time to Goal-IAPF	K_{att} GDA	K_{rep} GDA	Time to Goal-GDA
0.03	0.1	4.7s	1/300	600	7.48s
0.04	0.1	3.51s	1/320	600	7.52s
0.05	0.1	2.79s	1/340	600	7.56s

times of 7.48s to 7.55s, underlining its sensitivity to discrete updates.

The 4 trained models are tested in a real-world scenario see Fig. 7 with the UR5 as task robot (R1), human worker (H) and OpenManipulatorX as helper robot (R2) as the obstacles for 50 episodes each. Random goal position is created in every episode and real-time dynamic obstacles are tracked by the camera. Fig. 6 (a-d) gives the comparison between the simulation and the experimental results. In simulation, all different environments give similar performance for distance to goal and goal completion rates Fig. 6(a-b), but in experimental results, the environment with more observation performed less than the environment with more observation irrespective of their action space Fig. 6(c-d). The limitation of the field of view of the camera and the presence of dynamically moving obstacles in the workspace created a deviation in results in simulation and real-world experiments. It was evident that the environments with smaller observation space performed much better.The E3 environment with the small observation space and a large action space produced the best results Fig. 5 by reaching an average distance of 73mm from the goal with a goal completion rate of 0.866 where as E4 reached the goal at 128mm with a completion rate of 0.77, E1 has reached the goal at a distance of 204mm with a goal completion rate of 0.623 and E2 has reached the goal at 293mm with a completion rate of 0.467. As the aim was to provide a collision-free path, Fig. 6(a-f) show the closest distance reached from the obstacles in different

TABLE 6. Varying K(repulsive).

K_{att} IAPF	K_{rep} IAPF	Time to Goal-IAPF	K_{att} GDA	K_{rep} GDA	Time to Goal-GDA
0.04	0.1	3.51s	1/300	600	7.48s
0.04	0.2	3.51s	1/300	650	7.51s
0.04	0.3	3.52s	1/300	700	7.55s



(a) (b) (c)

(d) (e) (f)

FIGURE 6. (a-b) Experimental results for distance and goal completion rate, (c-f) Experimental results for different cases.

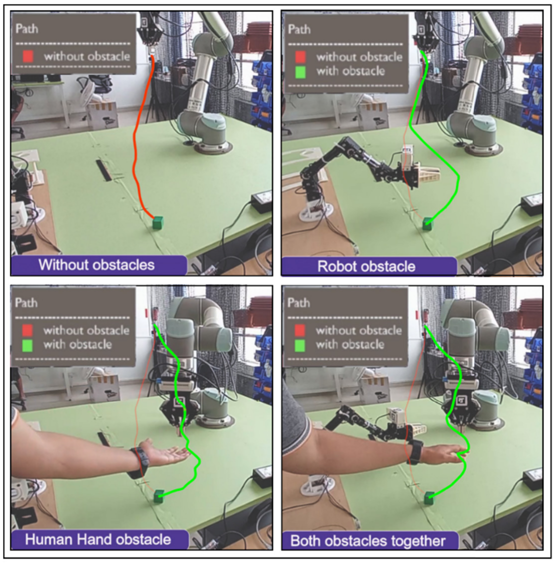


FIGURE 7. Comparison of path planned by DQN algorithm under dynamic obstacles in real-world experiment scenario.

observation space. It was observed that the trained models successfully avoided collisions in real-world experiments. The model with small observation and large action space produced the best results and is implemented in the final experimental setup with all the modalities working together. The camera simultaneously reads the environment states and RL model plans a safe action after each step for the UR5 to reach the goal without any collision and joint trajectory planning helps to smoothen the motion between each action. But since the trained model in simulation could reach till a distance on 74mm on average from the goal point, thus to attain better accuracy along with the safety, a goal proximity zone is created around the goal position. The proximity zone is a spherical volume of radius 80mm, encircling the goal point. The UR5 uses the safe path planned by DQN model to reach the proximity zone while avoiding collision. Upon reaching the goal proximity zone, the robot uses the point-to-point joint trajectory control to cover the last 80mm and reach the robot.

V. CONCLUSION

A new method has been developed for the collaboration of multi-robot and human operator in a shared workspace, which is bio inspired by the collaborative behavior of moray eels and coral groupers. Gestures and voice based commands are used for the communication between the 3 agents. A logistic regression model is used to train the two gesture classifier models, one for each robot, and `SSD_ResNet` model is used to train the object classifier model. A DQN model is trained to identify keywords from voice commands of the human operator. For obstacle avoidance, a Deep Q Network reinforcement learning approach is used to train the model on 4 different virtual environments and the results are then compared. It is observed that the environment with reduced observation space but larger action space produced the best results with the average final distance of 74.0mm from the goal position. The trained model is able to avoid obstacles and reach the goal in real world test scenarios as shown in Fig. 7. Also on comparing with 3 different Algorithm (IAPF, GDA, RRT*) its observed that IAPF outperformed in time taken to reach goal in all the 4 environments Fig. 4(q). Also, while comparing in terms of goal completion time IAPF performed with a median time of 4.7s while GDA performed with a median time of 7.62s and RRT* with 6.22s in different scenarios. The proposed approach is then compared with the trajectory path planning with Bezier curve interpolation approach without reinforcement learning. Although joint trajectory path planning provided a significantly better result in achieving the goal, it caused multiple collisions with obstacles. Thus, since the primary objective of our approach was to ensure safety by avoiding collision, the proposed approach successfully avoided any collision in the path and also ensured an intuitive collaboration between human operator and two robots in a shared workspace.

Future work includes a comprehensive comparative analysis of different reinforcement learning models, beyond DQN, will be conducted to identify optimal architectures for this specific task. Furthermore, a quantitative evaluation of the smoothness of the path will be incorporated, potentially using metrics such as curvature and jerk, to ensure more natural and efficient robot movements. A detailed analysis of the computational complexity of the proposed approach will be performed, considering factors such as training time and real-time execution speed, to assess its scalability for larger and more complex environments.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of the Robotics and Autonomous Systems Group, Brunel University, and the Robert Bosch Center for Cyber Physical Systems, IISc Bengaluru. They would like to thank Yu She, Mark Plecnik, Roel Pieters, G. Pisanelli, Xuesu Xiao, and M. Mitra for key discussions on the robot CAE, hri dynamics design, controller and motion planning improvements of the robot. They also would like to thank Pranav Rajesh Nair (IISc) and Raghu G. N. (IISc) for their valuable editing, review, and refinement of the article. They also like to acknowledge useful suggestions and feedback of the project given by G. Herzinger and Barbara Schilling (TU Munich) and Averil Horton (Brunel).

REFERENCES

[1] Q. Gao, Y. Chen, Z. Ju, and Y. Liang, "Dynamic hand gesture recognition based on 3D hand pose estimation for human–robot interaction," *IEEE Sensors J.*, vol. 22, no. 18, pp. 17421–17430, Sep. 2022.

[2] A. Jackowski, M. Gebhard, and R. Thietje, "Head motion and head gesture-based robot control: A usability study," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 1, pp. 161–170, Jan. 2018.

[3] J. Ondras, O. Celiktutan, P. Bremner, and H. Gunes, "Audio-driven robot upper-body motion synthesis," *IEEE Trans. Cybern.*, vol. 51, no. 11, pp. 5445–5454, Nov. 2021.

[4] P. P. Rao and A. R. Chowdhury, "Learning to listen and move: An implementation of audio-aware mobile robot navigation in complex indoor environment," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 3699–3705.

[5] J. Zurn and W. Burgard, "Self-supervised moving vehicle detection from audio-visual cues," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7415–7422, Jul. 2022.

[6] J. Chen and W.-J. Kim, "A human-following mobile robot providing natural and universal interfaces for control with wireless electronic devices," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 5, pp. 2377–2385, Oct. 2019.

[7] X. Hu, A. Song, H. Zeng, and D. Chen, "Intuitive environmental perception assistance for blind amputees using spatial audio rendering," *IEEE Trans. Med. Robot. Bionics*, vol. 4, no. 1, pp. 274–284, Feb. 2022.

[8] B. Beyret, A. Shafti, and A. A. Faisal, "Dot-to-dot: Explainable hierarchical reinforcement learning for robotic manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 5014–5019.

[9] S. Luo, H. Kasaei, and L. Schomaker, "Accelerating reinforcement learning for reaching using continuous curriculum learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[10] R. Jiang, Z. Wang, B. He, and Z. Di, "Vision-based deep reinforcement learning for UR5 robot motion control," in *Proc. IEEE Int. Conf. Consum. Electron. Comput. Eng. (ICCECE)*, Jan. 2021, pp. 246–250.

[11] Y. Ouyang, L. Dong, and C. Sun, "Critic learning-based control for robotic manipulators with prescribed constraints," *IEEE Trans. Cybern.*, vol. 52, no. 4, pp. 2274–2283, Apr. 2022.

[12] G. Onori, A. A. Shahid, F. Braghin, and L. Roveda, "Adaptive optimization of hyper-parameters for robotic manipulation through evolutionary reinforcement learning," *J. Intell. Robotic Syst.*, vol. 110, no. 3, p. 108, Jul. 2024.

[13] W. Li, J. Yue, M. Shi, B. Lin, and K. Qin, "Neural network-based dynamic target enclosing control for uncertain nonlinear multi-agent systems over signed networks," *Neural Netw.*, vol. 184, Apr. 2025, Art. no. 107057.

[14] L. Antonyshyn and S. Givigi, "Deep model-based reinforcement learning for predictive control of robotic systems with dense and sparse rewards," *J. Intell. Robotic Syst.*, vol. 110, no. 3, p. 100, Jul. 2024.

[15] A. R. Chowdhury, G. S. Soh, S. H. Foong, and K. L. Wood, "Experiments in robust path following control of a rolling and spinning robot on outdoor surfaces," *Robot. Auto. Syst.*, vol. 106, pp. 140–151, Aug. 2018.

[16] A. Palleschi, M. Hamad, S. Abdolshah, M. Garabini, S. Haddadin, and L. Pallottino, "Fast and safe trajectory planning: Solving the cobot performance/safety trade-off in human–robot shared environments," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5445–5452, Jul. 2021.

[17] T. Okudo and S. Yamada, "Learning potential in subgoal-based reward shaping," *IEEE Access*, vol. 11, pp. 17116–17137, 2023.

[18] M. Zhao and X. Lv, "Improved manipulator obstacle avoidance path planning based on potential field method," *J. Robot.*, vol. 2020, pp. 1–12, Jan. 2020.

[19] Y. Wang, Y. Sheng, J. Wang, and W. Zhang, "Optimal collision-free robot trajectory generation based on time series prediction of human motion," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 226–233, Jan. 2018.

[20] A. Pupa, M. Arrfou, G. Andreoni, and C. Secchi, "A safety-aware kinodynamic architecture for human–robot collaboration," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4465–4471, Jul. 2021.

[21] X. Cheng and S. Liu, "Dynamic obstacle avoidance algorithm for robot arm based on deep reinforcement learning," in *Proc. IEEE 11th Data Driven Control Learn. Syst. Conf. (DDCLS)*, Aug. 2022, pp. 1136–1141.

[22] R. Bshary, A. Hohner, K. Ait-El-Djoudi, and H. Fricke, "Interspecific communicative and coordinated hunting between groupers and giant moray eels in the red sea," *PLoS Biol.*, vol. 4, no. 12, p. e431, Dec. 2006.

[23] K. Joshi and A. Roy Chowdhury, "Bio-inspired vision and gesture-based robot-robot interaction for human-cooperative package delivery," *Frontiers Robot. AI*, vol. 9, Jul. 2022, Art. no. 915884.

[24] W. Wu, M. Chen, J. Li, B. Liu, and X. Zheng, "An extended social force model via pedestrian heterogeneity affecting the self-driven force," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7974–7986, Jul. 2022.

[25] M. Lippi and A. Marino, "Human multi-robot safe interaction: A trajectory scaling approach based on safety assessment," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 4, pp. 1565–1580, Jul. 2021.

[26] E. Coronado, T. Kiyokawa, G. A. G. Ricardez, I. G. Ramirez-Alpizar, G. Venture, and N. Yamanobe, "Evaluating quality in human–robot interaction: A systematic search and classification of performance and human-centered factors, measures and metrics towards an industry 5.0," *J. Manuf. Syst.*, vol. 63, pp. 392–410, Apr. 2022.

[27] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 5, no. 1, pp. 411–444, May 2022.

[28] D. Perille, A. Truong, X. Xiao, and P. Stone, "Benchmarking metric ground navigation," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Nov. 2020, pp. 116–121.

[29] H. Cheng, Y. Wang, and M. Q.-H. Meng, "A vision-based robot grasping system," *IEEE Sensors J.*, vol. 22, no. 10, pp. 9610–9620, May 2022.

[30] F. Piltan and S. Tayebi Haghighi, "Design gradient descent optimal sliding mode control of continuum robots," *IAES Int. J. Robot. Autom. (IJRA)*, vol. 1, no. 4, p. 175, Dec. 2012.

[31] X. Gao, H. Wu, L. Zhai, H. Sun, Q. Jia, Y. Wang, and L. Wu, "A rapidly exploring random tree optimization algorithm for space robotic manipulators guided by obstacle avoidance independent potential field," *Int. J. Adv. Robotic Syst.*, vol. 15, no. 3, May 2018, Art. no. 1729881418782240.

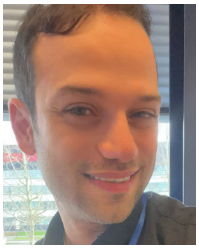
[32] Y. Hu, Y. Liu, A. Kaushik, C. Masouros, and J. S. Thompson, "Timely data collection for UAV-based IoT networks: A deep reinforcement learning approach," *IEEE Sensors J.*, vol. 23, no. 11, pp. 12295–12308, Jun. 2023.

[33] C. Scoccia, B. Ubezio, G. Palmieri, M. Rathmair, and M. Hofbaur, "Experimental assessment of a vision-based obstacle avoidance strategy for robot manipulators: Off-line trajectory planning and on-line motion control," *J. Intell. Robotic Syst.*, vol. 110, no. 3, p. 107, Jul. 2024.

[34] D. Fan, X. Zhang, and C. Wen, "Exponential regulation of uncertain nonlinear triangular impulsive systems: A logic-based switching gain approach," *IEEE Trans. Autom. Control*, vol. 70, no. 7, pp. 4881–4888, Jul. 2025.

[35] Y. Peng, Z. Yan, H. Zheng, and J. Guo, "Real time robot path planning method based on improved artificial potential field method," in *Proc. 37th Chin. Control Conf. (CCC)*, 2018, pp. 4814–4820.

JAYESH PRAKASH received the B.Tech. degree in mechanical engineering from the North Eastern Regional Institute of Science and Technology, Arunachal Pradesh, India, in 2019, and the M.Tech. degree in automation and robotics from the Defense Institute of Advanced Technology, Pune, India, in 2023. He has worked as a Research Assistant with the Robotics Innovations Laboratory (RIL), Indian Institute of Science (IISc), Bangalore. He is currently a Systems Engineer with Tata Consultancy Services Ltd., working in the field of advanced driver assistance systems (ADAS). His research interests include human–robot collaborations and multi-modal communications involving audio and visual cues.



SAM ALTNJI received the master's and Ph.D. degrees in biomechanics, in France.

He has a background in materials science engineering. He is currently an Assistant Professor of mechanical and industrial engineering with the Department of Design, Production and Management, University of Twente, The Netherlands,. His teaching and research interests include biomechanics, smart manufacturing, sustainable energy, and AI technologies, such as machine learning and digital twins. His research integrates materials science, structural mechanics, and numerical modeling, with applications in energy systems, biomechanical design, and intelligent manufacturing. He has contributed to the advancement of biomechanical systems, the enhancement of solar energy collectors using hybrid nanofluids, and the development of simulation-driven optimization frameworks for complex engineering processes. He also holds two European teaching qualifications UTQ (Netherlands) and a national teaching qualification from France.

KARTHICK THIYAGARAJAN (Senior Member, IEEE) received the B.E. degree in electronics and instrumentation engineering from Anna University, Chennai, India, in 2011, the M.Sc. degree in mechatronics from Newcastle University, Newcastle upon Tyne, U.K., in 2013, and the Ph.D. degree in smart sensing from the University of Technology Sydney (UTS), Sydney, Australia, in 2018.

He is currently a Lecturer of mechatronic and robotic engineering, the Director of the Smart Sensing and Robotics (SensR) Laboratory, the Assistant Stream Leader in Automation and Robotics at the Centre for Advanced Manufacturing Technology, and a Subject Coordinator for several mechatronics and robotics subjects at Western Sydney University. His research interests include novel sensing, perception and physical intelligence for applications in field robotics, inclusive healthcare robotics, advanced manufacturing and environmental monitoring.

Dr. Thiyagarajan was a Secretary of the IEEE Sensors Council NSW Chapter (2022–2025) and an Executive Committee Member of the IEEE Sensors Council Publicity Committee (2022–2025).

JOGESH S. NANDA received the bachelor's degree in electrical engineering from Kerala University, in 2014, and the master's degree in robotics and automation from Amrita Vishwa Vidyapeetham, in 2018. He is currently pursuing the Ph.D. degree with the Department of Design and Manufacturing, IISc, Bengaluru. Prior to the Ph.D. degree, he was the Team Lead Controls (Research and Development) at Hitech Robotics, where his work focused on multi-robot systems and system design for smart manufacturing and robotics solutions.

ABHIJIT BISWAS received the B.Tech. degree in electronics and instrumentation engineering from the University of Kalyani, in 2006, the M.E. degree in biomedical engineering from Jadavpur University, in 2008, and the Ph.D. degree in applied mechanics from IIT Madras, in 2015. He was a Postdoctoral Research Associate with MIT till 2018. He is currently an Assistant Professor with the Department of Design and Manufacturing, IISc. In his industrial career, he led Research and Development of Robhatah Robotic Solutions, Gears Energy Solutions, and Merkel Haptic Systems, and is a Founding Member of Dynocardia, Abhipsita Technologies, and Abhipsita Care.

ABHRA ROY CHOWDHURY (Senior Member, IEEE) received the Ph.D. degree in robotics and autonomous systems from the Department of Electrical and Computer Engineering, National University of Singapore, in 2015. He has been an Assistant Professor at the Department of Design and Manufacturing, Indian Institute of Science (IISc), Bengaluru. He was a Postdoctoral Research Fellow with Temasek Labs and Engineering Product Development (EPD) Pillar, Singapore University of Technology and Design (SUTD), from 2016 to 2019. He was associated with LG Electronics India, as a Senior Design Engineer, from 2010 to 2011, and Hitachi Labs Ibaraki, Japan, as a Scientist-I, from 2015 to 2016. He is the Founder and the Head of the Robotics Innovations Laboratory, IISc. He is an Erasmus Fellow (EU) and a Brunel Fellow, London. He served as a Senior Member for the Executive Committee of the DST Government of India, IEEE CSS, and OES Societies. He has consistently published in well-known robotics journals, flagship conferences, and delivered several talks in top-tier IEEE international conferences in robotics and autonomous systems. He holds various accolades and awards, including Singapore Mark Design Award, the IEEE Industry Applications Society (IAS) Best PhD Thesis Award, the IEEE Oceanic Engineering Society (OES) Award, the IEEE-SICE (Japan) Research Award, the IEEE-ICROS (S. Korea) Best Student Research Award, the President of India IIT Gold Medal, and the LG Electronics Gold Medal.