Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 59-18 (2025) 289–294

Combined Bi-RRT and Q-Learning path-planning in collaborative environments

Martina Pelosi * **Bianca Grieco** ** **Andrea Maria Zanchettin** ** **Paolo Rocco** **

* *Politecnico di Torino, Dipartimento di Automatica e Informatica (DAUIN), Corso Castelfidardo 34/d, 10138, Torino (Italy) (e-mail: martina.pelosi@polito.it)*

** *Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Piazza Leonardo da Vinci 32, 20133, Milano (Italy)*

Abstract: In recent years, a significant transformation towards intelligent manufacturing systems has been observed in industry. One of the leading research topics in this field is collaborative robotics, which promotes a synergic interaction between humans and robots. Advantages in ergonomics and production are foreseen with the adoption of collaborative robotics. Avoiding unintended collisions, which would ensure seamless collaboration, is one of the main challenges in improving safety and productivity. This paper focuses on a decision-making strategy that allows the robot to autonomously identify the optimal path to minimize the travel distance between the current configuration and the target while maintaining a safe distance from the human collaborator. The proposed strategy involves the offline generation of a dataset of possible paths within the robot workspace and a Reinforcement Learning-based control strategy, enabling the optimal choice of the subsequent robot configuration. After training and testing in a simulated environment, the optimal policy was validated with an ABB GoFa™ robotic arm, testing different human configurations and paths.

Keywords: Collaborative robots; Human-robotics interaction; Robot decision-making; Offline path generation; Reinforcement Learning; Robot control.

1. INTRODUCTION

The fourth industrial revolution marks a significant transformation towards flexible and intelligent manufacturing. Collaborative robotics is an essential component of Industry 4.0, enhancing production flexibility and human ergonomics by exploiting the complementary capabilities of humans and robots. They include human intelligence and dexterity to perform challenging tasks for robots and, on the other hand, robotic precision, strength, and repeatability to relieve humans from physically or mentally demanding operations (Chowdhury, 2023), (Patil et al., 2023). Despite the great potential of Human-Robot Collaboration (HRC), ensuring a seamless collaboration by avoiding collisions between humans and robots is one of the key challenges to improving the productivity of collaborative systems. Moreover, at the same time, a safe environment for the operators, compliant to the safety standards, must be enforced (Li et al., 2024). The research focuses on developing control laws and techniques to meet the required performance in collision avoidance, efficiency improvement, and production flexibility by designing robotic systems that can work safely and effectively alongside human workers promptly reacting to human behavior (Villani et al., 2018).

This paper falls within this research area, developing a decision-making strategy that allows the robot to choose the motion to perform based on the current human position. The approach aims to empower the robot to autonomously identify the optimal path both reducing task completion time and ensuring a safe distance from the worker. A database of admissible paths for a predefined robot task is created offline in a simulated environment using a Bidirectional-Rapidly Exploring Random Trees (Bi-RRT*) algorithm to enable the robotic arm to move from its starting position to the target one while avoiding static obstacles. This algorithm ensures comprehensive coverage of the workspace by concurrently expanding two trees from opposite ends and facilitates the connections at multiple points, optimizing the path-finding process. Subsequently, a Q-Learning (QL)-based method is developed to enable the robot to dynamically choose which of the previously computed paths to follow and when to transition from one path to another, depending on the human worker's presence. The training and testing phases of the QL algorithm exploit different sets of Motion Capture (MoCap) data, simulating human movements across different tasks. After training, an optimal policy is derived and subsequently validated on the ABB GoFa™ CRB 15000 collaborative robot.

Compared to the state-of-the-art methods, the key advantage of this research is the combination of a sampling-

10.1016/j.ifacol.2025.10.235

based strategy for multiple paths offline generation and a decision-making approach for collision avoidance, leading to the following main contributions:

- **Reduced path-planning computational effort**: the Bi-RRT* algorithm is used offline to generate a set of collision-free paths for the robot, considering only static obstacles without human presence. This precomputed graph ensures a well-connected set of feasible robot paths, among which the robot can choose during the execution, eliminating the need for costly real-time path generation.
- **Adaptability to generic human motion**: the QL-based strategy enables the robot motion to adapt to different human tasks by training the optimal policy on a generic dataset of randomly selected human motions. This strategy ensures a flexible and scalable policy that dynamically selects the optimal path based on real-time observations of human movement.
- **Task execution efficiency ensuring collision avoidance**: thanks to the dual value reward function, the length of the robot path is minimized while always ensuring collision avoidance with the operator.

2. RELATED WORKS

The development of control algorithms for collaborative robots (cobots) has advanced significantly to balance safety and task efficiency in shared human-robot workspaces.

Various approaches have been explored in path generation and path planning fields for collision avoidance purposes. Traditional methods, such as robot teaching and offline programming, are time-consuming, error-prone, and lack flexibility, which makes them impractical for modern industrial applications (Weber et al., 2023). Furthermore, in human-robot collaborative systems, the unpredictability of human actions increases the need for robust and scalable control strategies to enhance robot adaptability to dynamic environmental changes, disturbances, and uncertainties (Mazhar et al., 2023). Evolutionary computation techniques, such as Genetic Algorithms, Particle Swarm Optimization, and Differential Evolution, offer promising perspectives for handling complex optimization problems (Juříček et al., 2023). However, their real-world applicability is still challenging due to the extensive computational effort required by these approaches. Moreover, optimization problems are characterized by task-specific constraints, making evolutionary techniques more suitable for static environments. Optimization-based approaches, such as quadratic programming, are adaptive to environmental changes, but real-time applications still reveal high computational costs (Li et al., 2024).

Offline trajectory generation can improve path planning computational efficiency by pre-computing feasible paths assuming static obstacles and adjusting them online to fit real dynamic environments. For example, Scoccia et al. (2021) combines Artificial Potential Field (APF) and online motion control to adapt the precomputed trajectories to real obstacle motion. Similar approaches demonstrate their effectiveness also in collaborative setups. For instance, Tonola et al. (2021) proposes an offline path planner and a reactive re-planning framework to generate a new feasible path in case of human obstruction. However, re-planning is performed online, still requiring costly real-time computations. To solve this problem further improving the computational efficiency, Pellegrinelli et al. (2016) selects the optimal path for the actual context from a pre-generated database of paths avoiding areas frequently occupied by humans, based on precomputed human occupancy volumes and probabilities. In addition, sampling-based algorithms, such as Ant Colony Optimization (ACO) and Rapidly Exploring Random Tree (RRT/RRT*), have proved to produce optimal, efficient, and robust solutions in static environments and offline path-planning problems (Li et al., 2024). RRT* offers a good starting point for this paper, as it can generate offline a graph of feasible paths connecting a discrete set of configurations, randomly sampled by exploring the whole robot workspace. The extension of the sampled-based graph can be adjusted as needed without substantially affecting the efficiency of the online algorithm.

Decision-making algorithms can further improve path planning flexibility. In particular, reward-based approaches, like Reinforcement Learning (RL), offer strong adaptability to highly variable collaborative environments. RL is a robust method, proved to bring significant contributions in robot control for optimal operation choice (Shehawy et al., 2023) or for the manipulator's positioning and robot's path planning task (Lindner et al., 2021), (Zhao et al., 2021). Regarding Human-Robot Collaboration, some robot decision-making strategies rely on observed human actions and predicted motions (Jin et al., 2022) or human behavior in response to the robot's movements (Zhang et al., 2022). Recent RL-based strategies mainly focus on balancing human safety and task efficiency, but many rely on real-time computation, limiting their industrial feasibility. For example, Zhu et al. (2024) combines a safety field-based controller, modeling human and robot motions and task requirements, with Deep Reinforcement Learning (DRL) to optimize robot motion planning by balancing task efficiency with safety constraints. Liu et al. (2021) reports another example of motion optimization, introducing an Intrinsic Reward-Deep Deterministic Policy Gradient strategy to optimize policy learning. However, it was tested only in simulation. To improve computational efficiency while providing the required method scalability, Yu and Chang (2022) proposes a QL strategy exploiting a dataset of pre-demonstrated primitive actions to generate motion plans for new robot tasks and to request new demonstrations when needed.

Although, in our case, the robot task is fixed, human behavior can change, still requiring adaptation of the robot motion planning. A dataset of feasible and collision-free paths is pre-computed with Bidirectional-RRT* considering only fixed obstacles in the scene without human presence. A QL agent is then trained to choose the best transition from the current node to the next, among the available ones, allowing the adaptation of the robot motion to the observed worker behavior.

3. BI-RRT OFFLINE GENERATION OF A DATASET OF FEASIBLE ROBOT PATHS

This Section presents a method to generate a dataset of feasible, collision-free paths for a robotic manipulator operating in a workspace with static obstacles in the scene. The robot task involves moving the end-effector from a

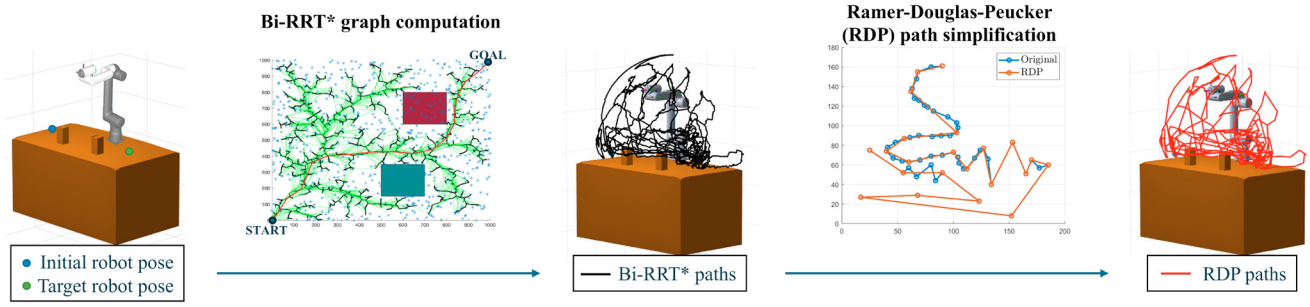


Fig. 1. Offline dataset generation of feasible paths: Bidirectional-Rapidly Exploring Random Trees (Bi-RRT) for rough path generation and Ramer-Douglas-Peucker (RDP) for path simplification

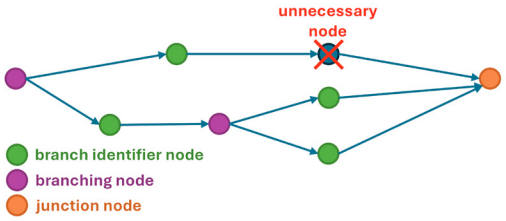


Fig. 2. Relevant nodes identification

predefined initial point to a target position. The goal is to create a graph of waypoints in the robot configuration space, eventually connecting the starting and final positions. This provides the robot with multiple motion choices during the task execution to avoid excessive proximity to the human operator. The Bi-RRT* algorithm is exploited to efficiently explore large areas of the space with a random sampling strategy, as shown in Figure 1. A comprehensive graph of possible paths in the workspace is generated by expanding two trees with root nodes at the specified start and goal configurations. With a reasonable selection of hyper-parameter values, such as maximum connection distance and iteration limit, it is possible to achieve a trade-off between planning time and path quality.

The resulting graph is refined by identifying only the feasible paths effectively leading to the goal configuration and removing dead ends. Moreover, the Ramer-Douglas-Peucker (RDP) algorithm is applied to simplify paths, as depicted in Figure 1. It reduces unnecessary nodes while preserving critical junctions and branching points. This process improves the robot's motion fluidity. Moreover, it simplifies the representation of states and actions, forming the basis to generate a Q-table to optimize online the robot's motion.

Finally, parent-child relationships between nodes are derived, removing nodes not filtered out by the RDP algorithm but unnecessary to unequivocally identify branching points, junction points, and single branches. Figure 2 further explains the distinction between relevant and unnecessary nodes. The final parent-child graph, consisting of N nodes and N_a connections between them, provides the hierarchical structure of the nodes within each feasible path. In this way, the available transitions from the current robot configuration to the possible next ones, i.e., the possible actions for the Q-Learning policy, are defined, enabling the implementation of the decision-making strategy. Within the N_a connections, self-loops for all nodes are added to indicate a waiting condition for the robot.

4. Q-LEARNING-BASED DECISION-MAKING STRATEGY

This Section describes the control strategy enabling the robot to select the optimal transition from the current node to the next one among those available in the pre-computed dataset of paths, minimizing the distance from the goal while avoiding collisions with the operator. A Q-Learning algorithm is employed for this purpose. Human activity is simulated using random motion samples from the Motion Capture (MoCap) dataset (Müller et al. (2007)), ensuring a realistic representation of human movements and a generalizable policy to unseen motions.

4.1 Environment and reward definition

To implement and train the optimal QL policy, the agent, the environment, and the reward must be defined. The manipulator is the only agent, while the human is part of the environment and affects its state. The workspace is discretized into N_v voxels (Figure 3), each represented by its center Cartesian coordinates relative to the robot reference frame and side length L. In this way, a structured and computationally manageable environment representation is created. To comprehensively describe the environment of the Markov Decision Process, each state is defined as:

$$S_t = (n_t, v_t)$$

with n_t the robot's current node and v_t the voxel occupied by the worker. To simplify the implementation, only the human-hand motion is considered as input, thus reducing the number of possible states. Moreover, a dummy voxel is introduced to account for cases where the human is outside the workspace, i.e. not occupying any of the N_v voxels. The total number of states is: $N_s = (N_v + 1) \cdot (N - 1) + 1$. The following reward function guides the learning process by balancing collision avoidance and path efficiency.

$$R_t = R_t^1 + R_t^2 \tag{1}$$

$$R_t^1 = \begin{cases} -1000 & \text{if } d_v(t) = 0, \\ 50 \cdot \ln\left(\dfrac{d_v(t)}{2a}\right) \cdot d_v(t) & \text{if } d_v(t) \leq a, \\ \ln\left(\dfrac{d_v(t)}{2a}\right) \cdot d_v(t) & \text{otherwise.} \end{cases} \tag{2}$$

$$R_t^2 = \mu \cdot \frac{1}{d_G(t)} \tag{3}$$

where $d_G(t)$ and $d_v(t)$ are the distances between the current robot node n_t and the target configuration and the

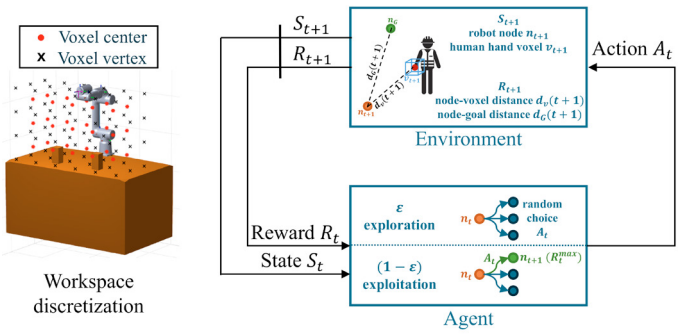
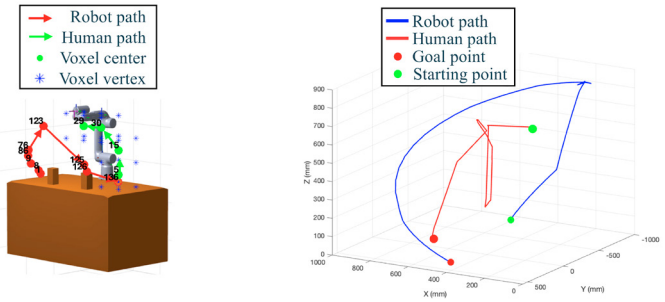


Fig. 3. Workspace discretization in N_v voxels and Q-Learning training process



(a) Offline policy testing: robot optimal path with random MoCap human motion

(b) Online validation: human and robot recorded paths

Fig. 4. Examples of offline and online test results: simulated and recorded robot paths

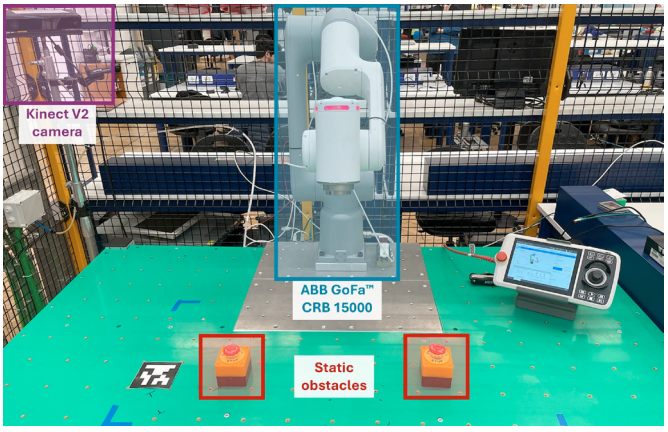


Fig. 5. Experimental setup comprising an ABB CRB15000 robot, a Microsoft Kinect V2 camera and two static obstacles

center of the voxel occupied by the human n_v, respectively; μ is an adjusting factor; $a = \frac{1}{2}\sqrt{3L^2}$ is a safety threshold, corresponding to the distance between the vertex of a voxel and its center. The total reward R_t is the sum of two contributions:

- The conservative term R_t^1 (equation 2) strongly penalizes close proximity between the robot's end-effector and the voxel occupied by the human worker (i.e., $d_v(t) \leq a$) with a maximum penalty of -1000 when colliding (i.e., $d_v(t) = 0$); the reward becomes positive, increasing with the distance $d_v(t)$, only in safe conditions (i.e., $d_v(t) > 2a$).
- The proactive term R_t^2 (equation 3) rewards shorter distances from the current robot configuration to the target node $d_G(t)$.

4.2 Policy training and testing

The QL algorithm, which is also trained offline and is comprehensively described in Figure 3, employs an ϵ-greedy strategy for action selection. Initially, exploration is prioritized ($\epsilon = 0.9$), gradually shifting towards exploitation in the second half of training episodes thanks to an exponentially decreasing ϵ-factor. Given the environment's complexity, two different strategies are followed to accelerate convergence and efficiently optimize the Q-table. First, as in MDPs the future state depends only on the current state (n_t, v_t) and action A_t (i.e. the transition from the current node to the next), the robot's starting position is randomly selected from the Bi-RRT graph for each episode. In addition, un-feasible transitions are not considered: during exploration, the action A_t is selected only within the set of feasible transitions at the current node, avoiding unnecessary computation.
After 50,000 training episodes, the Q-table converges, reaching the optimal policy. For 7% of states, the self-loop action is associated with the maximum reward, suggesting that all transitions to other nodes are undesirable. In fact, these actions show negative rewards, indicating that, although permitted, they lead to a robot node located within the same voxel occupied by the human or extremely close to it. The self-transition mechanism allows the robot to wait at the same node until the human voxel changes, altering the environment state and enabling the algorithm to discover a desirable transition based on a new combination of current node and new voxel occupied.
The optimal trained policy is tested on 10 new human MoCap tasks, not used for training. It successfully identifies the shortest sequence of nodes that avoids the voxels occupied for the specific human in 100% of cases. Figure 4a illustrates an example of the manipulator's path in response to a sequence of human hand-occupied voxels, randomly sourced from the MoCap dataset.

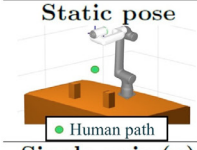
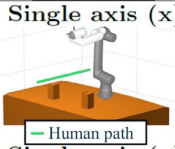
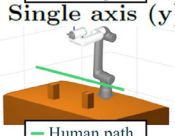
5. EXPERIMENTAL VALIDATION

All parameter values exploited for the implementation and offline training of the optimal policy are summarized in Table 1.
For the experimental validation, a Microsoft Kinect V2 sensor and the Mediapipe Hand Landmarker algorithm (Zhang et al., 2020) were used to track the human wrist's position. The experimental setup can be found in Figure 5. Some tests were conducted on an ABB GoFa™ robotic arm to validate the effectiveness of the method under various scenarios: 1 test without human presence ("No human"), 5 with human hand position fixed in the workspace ("Static

Table 1. Parameters values for the decision-making policy training

# nodes	# actions	# voxels	# states	Voxel side	Safety threshold	Adjusting factor
N	N_a	N_v	N_s	L	a	μ
131	331	30	4031	30	25.98	25

Table 2. Test results: task completion time T, human occupancy volume V, and minimum robot-human distance $d_{v_{min}}$

Test	ID	T [s]	V [cm^3]	$d_{v_{min}}$ [cm]
No human	-	18.48	-	-
Static pose	1	28.79	6.23	44.58
	2	18.48	7.93	52.88
	3	30.58	10.23	27.78
	4	39.48	17.28	36.17
	5	29.71	8.22	32.80
Single axis (x)	1	50.52	145.60	30.84
	2	32.70	252.31	54.76
	3	30.44	191.24	37.81
	4	39.28	269.56	34.98
	5	35.51	160.09	16.30
Single axis (y)	6	28.07	9.03	35.85
	7	29.00	196.64	13.34
	8	43.05	430.37	29.34
	9	42.58	686.46	44.57
	10	39.17	960.02	26.24
Single axis (z)	11	25.13	23.61	41.39
	12	34.11	67.91	49.44
	13	36.93	118.81	47.45
	14	86.30	68.90	55.69
	15	47.12	134.47	56.83
Random case	1	34.04	1147.55	31.08
	2	37.11	621.71	28.52
	3	35.85	416.96	27.49
	4	39.59	967.10	29.71
	5	22.06	49.54	36.29
	6	28.80	747.98	32.56
	7	29.75	563.02	29.15
	8	26.64	936.73	26.75
	9	32.08	499.96	31.94
	10	27.77	287.62	28.54

pose"), 15 with human hand moving linearly along one axis ("Single axis", five tests per direction), and 10 with human performing random task, i.e., 3D-hand motions ("Random motion").

5.1 Results and discussion

All tests confirmed collision-free operations. As an example, Figure 4b shows the recorded paths followed by the human hand and the robot end-effector over time for the third test of the "Random motion" category.

For a comprehensive algorithm validation, two distinct KPIs were used: the time needed for completing the task T, assessing the method's operational efficiency, and the minimum distance between the end-effector and the human hand $d_{v_{min}}$, verifying that the minimum safety distance a is kept throughout the entire task execution. In addition, the human occupancy volume V was computed by treating the sequence of point coordinates occupied by the human's hand over time as a point cloud volume. The recording of this metric aims to explore a possible correlation between task execution time T and the extent of human movement within the workspace.

The test results are presented in Table 2. The fastest time (18.48s) occurred in the "No human" test and in the second "Static pose" test: in these cases, the robot performed the shortest available path, as the human pose never interferred with it. The distance between the human hand and the robot's end-effector consistently exceeds the established safety threshold a in 93.3% of cases. Only two "Single axis" tests do not satisfy the threshold. The robot was moving toward a node considered safe based on its distance from the worker at the previous node. Still, the end-effector path came too close to the human hand one, as the human positions were approximated with the occupied voxel center. In addition, the subsequent robot node is chosen based on the current human hand location, not accounting for the upcoming human movement. The occupancy volume varies significantly, up to 1147.55cm^3. This parameter is lower in the "Static pose" tests, where the human is confined to a single voxel and reaches its peak in the "Random motion" group, as it reflects the extension and complexity of human movement within the workspace. The correlation value between this parameter and execution time T is 0.05, revealing almost no dependence. The task execution time is not influenced by the extent of the human motion in the workspace, i.e., whether the hand is stationary or in motion, but rather by the specific voxel occupied when the robot is asked to move.

In addition, Figure 6 compares the minimum robot-human distance $d_{v_{min}}$ and task completion time T across all tests. The overall correlation coefficient between the two KPIs is 0.28, suggesting a moderate positive dependence. However, the plot highlights an outlier corresponding to the fourteenth "Single axis" test, significantly increasing the correlation value. This point has a substantially longer completion time ($T = 86.30s$), although the minimum reached distance d_{vmin} more than double the safety threshold a. During the test generating the outlier, the robot's path is subject to some pauses: the best action corresponds to the self-transition, as all the other possible moves are deemed too close to the human. This safety strategy greatly extends the execution time but avoids stopping and restarting the robot operation, allowing a seamless collaboration. Without this outlier, the correlation value decreases to almost 0 ($\simeq$ -0.0007), indicating that the robot deviations from the shortest possible path, imposed to avoid collisions when the manipulator is closer to the operator, have almost no impact on the execution efficiency. This result is achieved thanks to the multiple alternative efficient routes present in the Bi-RRT* path network, allowing the robot to maintain safety without significantly increasing execution time.

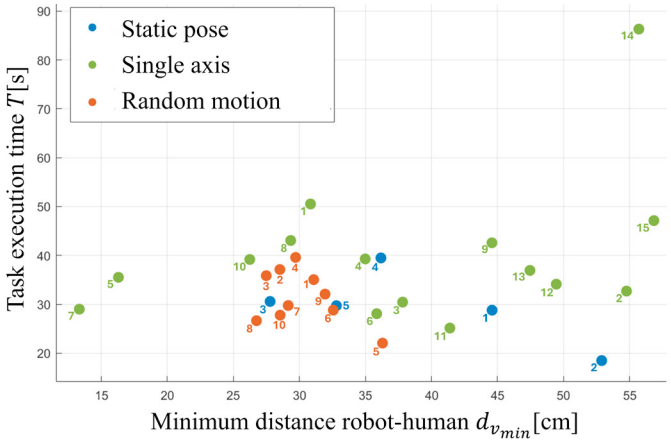


Fig. 6. Task execution time T against minimum robot-human distance $d_{v_{min}}$

6. CONCLUSIONS

In collaborative robotics, production efficiency, flexibility, and human safety are critical concerns. The developed Q-Learning strategy enables robots to detect and react to human movements, preventing collisions while ensuring seamless collaboration and enhancing productivity. A key strength of the proposed approach is its generalizability to various human tasks and applications, which was proved by validating the trained policy to unseen human paths. Furthermore, offline path generation and policy training eliminate the need for complex real-time calculations, enhancing practical feasibility. Experimental validation on the ABB GoFa™ robotic arm confirmed the algorithm's effectiveness in maintaining a safe distance from the operator's hand. Despite these achievements, the approach shows some limitations, including occasional breaches of the safety threshold and robot stops increasing the execution time. These problems are due to insufficient node density or imprecise human pose estimation. Enhancing the node number and the workspace discretization could help avoid these critical conditions. Moreover, incorporating a predictive strategy for human motion estimation could enhance the decision-making process, improving the method's accuracy and reliability.

ACKNOWLEDGEMENTS

This study was partially carried out within the MICS (Made in Italy – Circular and Sustainable) Extended Partnership and received funding from Next-Generation EU (Italian PNRR – M4 C2, Invest 1.3 – D.D. 1551.11-10 2022, PE00000004). CUP MICS D43C22003120001

REFERENCES

Chowdhury, H. (2023). Human-robot collaboration in manufacturing assembly tasks. *Preprints*. doi:10.20944/preprints202310.0049.v2. URL `https://doi.org/10.20944/preprints202310.0049.v2`.

Jin, Z., Liu, A., Zhang, W.A., Yu, L., and Su, C.Y. (2022). A learning based hierarchical control framework for human–robot collaboration. *IEEE Transactions on Automation Science and Engineering*, 20(1), 506–517.

Juříček, M., Parák, R., and Kůdela, J. (2023). Evolutionary computation techniques for path planning problems in industrial robotics: A state-of-the-art review. *Computation*, 11(12). doi:10.3390/computation11120245. URL `https://www.mdpi.com/2079-3197/11/12/245`.

Li, W., Hu, Y., Zhou, Y., and Pham, D.T. (2024). Safe human–robot collaboration for industrial settings: a survey. *Journal of Intelligent Manufacturing*, 35(5), 2235–2261.

Lindner, T., Milecki, A., and Wyrwał, D. (2021). Positioning of the robotic arm using different reinforcement learning algorithms. *International Journal of Control, Automation and Systems*, 19, 1661–1676.

Liu, Q., Liu, Z., Xiong, B., Xu, W., and Liu, Y. (2021). Deep reinforcement learning-based safe interaction for industrial human-robot collaboration using intrinsic reward function. *Advanced Engineering Informatics*, 49, 101360.

Mazhar, A., Tanveer, A., Izhan, M., and Khan, M.Z.T. (2023). Robust control approaches and trajectory planning strategies for industrial robotic manipulators in the era of industry 4.0: A comprehensive review. *Engineering Proceedings*, 56(1), 75.

Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., and Weber, A. (2007). Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn.

Patil, S., Vasu, V., and Srinadh, K. (2023). Advances and perspectives in collaborative robotics: a review of key technologies and emerging trends. *Discover Mechanical Engineering*, 2(1), 13.

Pellegrinelli, S., Moro, F.L., Pedrocchi, N., Tosatti, L.M., and Tolio, T. (2016). A probabilistic approach to workspace sharing for human–robot cooperation in assembly tasks. *CIRP Annals*, 65(1), 57–60.

Scoccia, C., Palmieri, G., Palpacelli, M.C., and Callegari, M. (2021). A collision avoidance strategy for redundant manipulators in dynamically variable environments: online perturbations of off-line generated trajectories. *Machines*, 9(2), 30.

Shehawy, H., Pareyson, D., Caruso, V., De Bernardi, S., Zanchettin, A.M., and Rocco, P. (2023). Flattening and folding towels with a single-arm robot based on reinforcement learning. *Robotics and Autonomous Systems*, 169, 104506.

Tonola, C., Faroni, M., Pedrocchi, N., and Beschi, M. (2021). Anytime informed path re-planning and optimization for human-robot collaboration. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 997–1002. IEEE.

Villani, V., Pini, F., Leali, F., and Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55, 248–266.

Weber, A.M., Gambao, E., and Brunete, A. (2023). A survey on autonomous offline path generation for robot-assisted spraying applications. *Actuators*, 12(11). doi:10.3390/act12110403. URL `https://www.mdpi.com/2076-0825/12/11/403`.

Yu, T. and Chang, Q. (2022). Motion planning for human-robot collaboration based on reinforcement learning. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 1866–1871. IEEE.

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.L., and Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*.

Zhang, R., Li, X., Zheng, Y., Lv, J., Li, J., Zheng, P., and Bao, J. (2022). Cognition-driven robot decision making method in human-robot collaboration environment. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 54–59. IEEE.

Zhao, Y., Zhang, Y., and Wang, S. (2021). A review of mobile robot path planning based on deep reinforcement learning algorithm. In *Journal of Physics: Conference Series*, volume 2138, 012011. IOP Publishing.

Zhu, C., Yu, T., and Chang, Q. (2024). Task-oriented safety field for robot control in human-robot collaborative assembly based on residual learning. *Expert Systems With Applications*, 238, 121946.