applied sciences

MDPI

Article

Goal-Conditioned Reinforcement Learning within a Human-Robot Disassembly Environment

Íñigo Elguea-Aguinaco [1,2], Antonio Serrano-Muñoz [2], Dimitrios Chrysostomou [3], Ibai Inziarte-Hidalgo [1], Simon Bøgh [3] and Nestor Arana-Arexolaleiba [2,3,*]

[1] Research & Development Department, Electrotécnica Alavesa S.L., 1010 Vitoria-Gasteiz, Spain
[2] Robotics and Automation Electronics and Computer Science Department, University of Mondragon, 20500 Mondragon, Spain
[3] Materials and Production Department, Aalborg University, 9220 Aalborg East, Denmark
* Correspondence: narana@mondragon.edu, naar@mp.aau.dk; Tel.: +34-943-794-700

Abstract: The introduction of collaborative robots in industrial environments reinforces the need to provide these robots with better cognition to accomplish their tasks while fostering worker safety without entering into safety shutdowns that reduce workflow and production times. This paper presents a novel strategy that combines the execution of contact-rich tasks, namely disassembly, with real-time collision avoidance through machine learning for safe human-robot interaction. Specifically, a goal-conditioned reinforcement learning approach is proposed, in which the removal direction of a peg, of varying friction, tolerance, and orientation, is subject to the location of a human collaborator with respect to a 7-degree-of-freedom manipulator at each time step. For this purpose, the suitability of three state-of-the-art actor-critic algorithms is evaluated, and results from simulation and real-world experiments are presented. In reality, the policy's deployment is achieved through a new scalable multi-control framework that allows a direct transfer of the control policy to the robot and reduces response times. The results show the effectiveness, generalization, and transferability of the proposed approach with two collaborative robots against static and dynamic obstacles, leveraging the set of available solutions in non-monotonic tasks to avoid a potential collision with the human worker.

Keywords: collaborative robots; machine learning; reinforcement learning; contact-rich tasks; disassembly; collision avoidance

Citation: Elguea-Aguinaco, Í.; Serrano-Muñoz, A.; Chrysostomou, D.; Inziarte-Hidalgo, I.; Bøgh, S.; Arana-Arexolaleiba, N. Goal-Conditioned Reinforcement Learning within a Human-Robot Disassembly Environment. *Appl. Sci.* **2022**, *12*, 11610. https://doi.org/10.3390/app122211610

Academic Editor: Emanuele Carpanzano

Received: 3 October 2022
Accepted: 11 November 2022
Published: 15 November 2022

1. Introduction

In recent decades, technological advances and digitalization have driven economic growth and improved everyday life in countless ways. However, the increasing reliance on electronic products has caused an abundance of electronic waste. In fact, waste electrical and electronic equipment (WEEE) represents the fastest growing type of waste in Europe [1], despite policies adopted by several countries that oblige manufacturers to take back these products from customers for treatment [2].

Notwithstanding, remanufacturing seems to shed light on this environmental concern through disassembly, which allows the recovery of valuable components and materials from these wastes for future reuse. While human labor still predominates in this process, over time, many tasks have been automated through industrial robots that have improved the cost-effectiveness of companies. However, robots cannot fully replace human operators in many environments due to the high variability of the end-of-life (EoL) products received [3].

Therefore, human-robot interaction (HRI) offers a flexible solution that combines the capabilities of a collaborative robot and a human co-worker. Although its applicability is widening, the interaction between these two agents still causes productivity-reducing obstacles e.g., the use of safety stops can slow workflow to ensure operator safety. In HRI applications, such as disassembly plants, where the robot and operator share a workspace in

an unstructured environment, this drawback makes it important to improve these systems' perception and decision-making [4].

In this regard, artificial intelligence (AI) and, specifically, machine learning (ML) emerge as promising solutions. Indeed, there has been a growing interest in applying these techniques in multiple domains, such as healthcare [5], autonomous driving [6] or automation, and robotics [7], even in complex scenarios. In particular, reinforcement learning (RL) enables a robot to discover optimal behavior through trial-and-error interactions with its environment autonomously. However, conventional RL only addresses the case of a single goal, specified by a single reward function [8]. This approach leads to limitations such as long learning times, poor generalization, and performance drops when transferring a policy learned in simulation to a real robot when the agent must learn multiple goals. In addition, most current approaches use third-party libraries or communication tools, such as Robot Operating System (ROS), to send the RL policy action commands to the robot. This communication slows the robot's response time, compromising the co-worker's safety in highly dynamic environments. Consequently, these difficulties continue to limit RL's potential and its deployment in realistic applications. Goal-conditioned RL formalizes the problem of learning different goals in one environment [9]. Similar to conventional RL, the main difference in goal-conditioned RL is that, in this case, the reward function depends on the agent's goals, which are represented by the states of the system. This approach may lead to multiple sources of reward during learning. Most goal-oriented RL approaches proposed in the literature include the goal as part of the observation, understood as a final pose [10,11]. However, in this work, the goal is defined as task completion, giving the robot the freedom to select available trajectories.

Thus, this paper presents an approach that simultaneously tackles disassembly and collision avoidance with deep RL techniques in an HRI scenario in which the robot must extract a peg while considering the co-worker's position (Figure 1). This approach uses a model-free algorithm, avoiding modeling collision forces in disassembly or co-worker behavior. Specifically, the contributions of this paper are:

- A goal-conditioned RL approach that considers co-worker safety through real-time collision avoidance during the performance of non-monotonic disassembly tasks.
- Improved agent generalization capabilities to extract parts of different frictions, tolerances, and orientations, and despite noisy image captures.
- A new scalable multi-control framework for direct transfer of RL policies between robots and their control in reality independent of other software libraries.

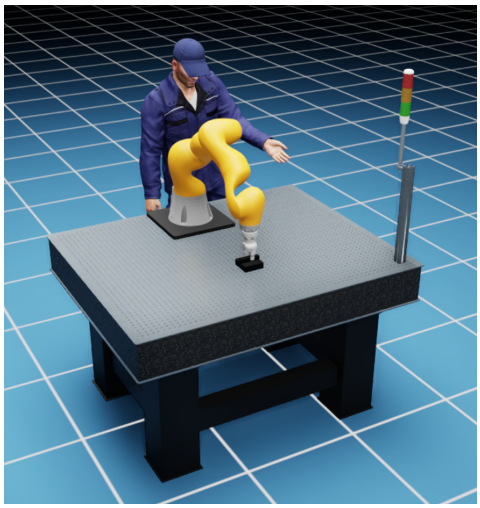

Figure 1. HRI scenario in which the robot performs a contact-rich disassembly by extracting a peg to the opposite side of the co-worker thereby avoiding a potential collision between the two agents.

To the authors' knowledge, the proposed approach is among the first examples that simultaneously apply only deep RL techniques in contact-rich manipulation tasks and collision avoidance.

The content of the paper is organized as follows. Section 2 contains a brief description of related works. In Section 3, a theoretical background on RL is provided and its basic concepts are defined for the particular study. Section 4 describes the task to be performed and presents the results obtained both in simulation and in reality and on different serial manipulators. Section 5 provides an outline of the main headings to consider for future research, providing points for discussion and reflection. Lastly, Section 6 concludes with a summary of the knowledge gained.

2. Related Works

2.1. Contact-Rich Manipulation Tasks: Assembly and Disassembly

The effective resolution of complex manipulation tasks in an unstructured or highly variable environment remains an active field of research. Current research focuses mainly on grasping [12], picking and placing [13], and assembly tasks [14]. In particular, RL methods have shown high robustness to uncertainties in the latter, leading more and more researchers to focus on learning assembly skills.

Currently, the application of RL in assembly is focused on three lines of research: improving performance, sample efficiency, and generalization capability and narrowing the simulation-reality gap. Performance improvement spans different domains although research generally focuses on accuracy [15], safety [16], robustness [17], and contact stability [18]. Other studies, such as [19,20], analyzed stability from a different perspective considering that any state trajectory must be bounded and tend to the target position required by the task. For this purpose, the authors shaped the exploration of the RL agent with a Lyapunov function. This constrained exploration, in turn, guaranteed the manipulator's safe and predictable behavior. However, this approach had limitations, such as the control policy being valid only for episodic tasks with a single goal position or that stability was preserved only in passive environments where the robot's behavior did not depend on human users. In addition, many of these studies that focused on improving the performance of RL policies in assembly reported long training periods [21]. This limitation is a recurrent drawback mainly in high-dimensional tasks requiring extensive exploration.

Therefore, many studies also put their efforts into mitigating this limitation and obtaining higher sampling efficiency. Two of the most explored lines of research in recent years to increase sample efficiency have been the combination of RL with human demonstrations [22] and meta-learning [23]. Lastly, generalization capability, as well as the deployment of the policy from simulation to reality, are two of the main challenges reported around RL at present. As such, many studies related to assembly focus their concern directly on this issue. Both concepts, generalization, and deployment of learned policy into reality, are closely related, as greater generalizability leads to robust policies that are less susceptible to the simulation-to-reality gap. In this sense, one of the most widely used approaches is based on domain randomization [24]. This technique allows uniformly randomizing a distribution of actual data in predefined ranges in each training episode to obtain more robust policies.

Nonetheless, unlike assembly, few papers in the literature deal with disassembly. For instance, Kristensen et al. [25] proposed a Q-learning algorithm framework to train and test agents in robotic unscrewing tasks. Simonič et al. [26] created a framework that mapped the information obtained during disassembly tasks to assembly tasks. They implemented a hierarchical RL algorithm and a graph representation under the criterion that disassembly is the reverse of assembly broken down into multiple stages. However, the proposed approach was only suitable for cases where the assembly task was reversible, which essentially limited it to standardized disassembles. In turn, Herold et al. [27] proposed strategies to separate fixed components into a slot. For this, the authors identified that adjusting the robot's position end-effector proportionally to the measured forces and including oscillating motion

could be a suitable solution for the assignment. Nevertheless, they executed the task by performing predefined actions, making it difficult to generalize to other scenarios. In this sense, Serrano-Muñoz et al. [28] specifically analyzed the generalization capability of two actor-critic algorithms in contact-rich disassembly tasks. For this purpose, they randomized both the rotation and the position of a peg embedded in a base that the robot had to extract, obtaining promising real-world results.

Indeed, generalization capability might play a key role in disassembly since, unlike in assembly, the condition of an EoL product can be highly heterogeneous. Therefore, disassembly through RL should consider agents dealing with the physical uncertainties associated with the product condition, considering the large variety within one product category, and complexities in process planning and operation.

2.2. Path Planning and Collision Avoidance

Collision avoidance of robotic manipulators is a hot topic in robot control research. This technique finds a sequence of joint configurations that allows the robot to move from an initial to a target point while avoiding potential obstacles along the way. Conventional path planning methods like rapid-exploring random trees (RRT) [29] and artificial potential fields (APF) [30] are well-researched techniques that can compute a trajectory with a lower computational cost than RL. However, RRT is inefficient for fast, reactive, and dynamic collision avoidance action [31], and APF tends to fall into local minima when attractive and repulsive forces are similar.

Alternative solutions include predicting the motion or recognizing the operator's intentions. For instance, while Yasar and Iqbal [32] used recurrent neural networks to predict the operator's movement based on its position, velocity, and acceleration, Buerkle et al. [33] used an electroencephalogram to detect the operator's shift intentions before the movement was performed. However, although the results were promising, in practice, it remains challenging due to complex human behavior and background noise levels in many industrial plants.

In this sense, RL allows adaptive control in dynamic environments due to its self-learning capability when the working environment varies, making it an appealing solution for unstructured environments such as HRI scenarios. Most research relies on actor-critic algorithms [34]. For instance, Prianto et al. [35] proposed a path planning algorithm for a multi-arm manipulator which they considered as a single-arm virtual manipulator steering from an initial to a final configuration. In a similar case study, Zhou et al. presented two-path planning methods using either residual RL [36] or curriculum RL [37] to find the axis configuration that allowed the robot to perform a collision-free trajectory. However, all these studies validated their methodology only with static obstacles.

In turn, El-Shamouty et al. [38] presented a framework that mapped HRI tasks and safety requirements onto RL settings. However, the generated human motion in the collision avoidance simulation was randomized without considering possible hazardous scenarios. The same limitation can be noticed in the work of Prianto et al. [39] and Sangiovanni et al. [40,41]. In the former, the obstacles moved in limited trajectories periodically, while in the latter a real-time model-free collision avoidance approach was introduced and applied to robotic tasks where an unpredictable obstacle interfered with the robot's workspace. Xiong et al. [42] proposed a real-time robotic manipulator motion planning framework which they tested with both static and dynamic obstacles. Nevertheless, the negative reward given to the agent was only provided after a certain threshold was exceeded, which could increase training time and even affect convergence if the target was not reached during exploration. To avoid a similar situation, Zhao et al. [43] included a balance estimator in the reward function that compared worker safety with task performance efficiency. In addition, in this study, the obstacle had a motion acquired from skeleton tracking. Yet, as in the other studies, the methodologies were evaluated in via-point experiments, where the robot only moved from an initial to a target point. This does not detract from

the fact that in HRI tasks, despite ensuring co-worker safety, the robot should keep the workflow efficient, and accomplish those tasks assigned to it.

In this regard, one of the few papers that addressed contact-rich manipulation tasks while the robot agent avoided collision with its surroundings is Yamada et al. [44]. However, although the framework integrated the motion planner into the RL policy, collision avoidance for static obstacles was performed by RRT, and RL completed contact-rich tasks, so the authors switched between the two techniques.

Unlike previous works, this study aims to evaluate whether the hypothesis "*an RL agent is capable of performing contact-rich disassembly tasks while avoiding potential collisions with dynamic human movements*" is valid. To this end, this paper proposes a model-free approach using entirely deep RL techniques, which identifies that suitable policy for contact-rich disassembly tasks in collision-free HRI environments. Having such a policy in place, we can emphasize co-worker safety during the interaction with the robot. To the best of our knowledge, the proposed method is the first model-free approach using deep RL techniques that considers co-workers' potential collisions during HRI. The use case is characterized by a dual solution whose target position depends on the obstacle location instead of a given initial and target position as in previous works. The robustness and performance of the algorithm are then tested considering task- and human-related variables, such as the position of the object to be disassembled in multiple rotations and under static and dynamic obstacles, respectively.

3. Method

3.1. Markov Decision Process

The RL framework [45] relies on an agent learning autonomously how to interact with its environment to perform a given task. This interaction between the agent and its environment is formalized as a Markov Decision Process (MDP), which defines sequential decision-making as a semi-random process according to which the agent acts. An MDP is usually represented by the following tuple:

$$[S, A, P(s_{t+1}|s_t, a_t), R(s_t, s_{t+1}, a_t), \gamma] \tag{1}$$

At any time step t, the agent observes its environment, represented by a given state $s_t \in S$, where S represents the state space. Based on its behavior, denoted as policy, $\pi(a|s)$, the agent will take an action $a_t \in A$, where A is the space of actions. Depending on the transition probability $P(s_{t+1}|s_t, a_t)$, the action a_t will lead the agent to a new state s_{t+1}, and it will receive a reward r_t, a scalar representing how good the agent's action was towards the task at hand. The agent's objective will be to maximize the cumulative rewards received over the long run, $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$, where γ is a predefined discount factor $0 \leq \gamma \leq 1$ determining the present value of future rewards. The total amount of reward that an agent expects to accumulate in the future, starting from a particular state s_t and following a particular policy $\pi(a|s)$, is called the value function.

3.1.1. State Space

The relevant observations for the training process are the position and rotation of the base where the peg is inserted in the x-y plane, bx, y; the position and rotation of the end-effector in the x-y plane, at the current time step, $ee_{x,y}$, as well as at the previous time step, $pee_{x,y}$; and the collision forces between the end-effector and the base in the three axes, $F_{x,y,z}$. The percentage of part extraction, p_{extr}, is also measured, and the number of timesteps completed is counted against the total, t_{ep}, which is later used in the reward to speed up the disassembly process. Concerning the position of the obstacle in the x-y plane, $obs_{x,y}$, it is assumed to be known and provided by the cameras located in the workspace. Thus, the state space S is defined as:

$$S = [b_{x,y}, ee_{x,y}, pee_{x,y}, F_{x,y,z}, p_{extr}, t_{ep}, obs_{x,y}], \tag{2}$$

3.1.2. Action Space

The choice of action space significantly impacts the robustness and performance of the learned policy. A common approach is to work in the joint space of the robot [46]. However, in this approach, each action requires approximating the Jacobian of each pose. In addition, erroneous Jacobian modeling may prevent an optimal transfer of learning to reality. Instead, working in task space allows sending the pose commands in Cartesian space to the robot's internal controller, which uses the internally encoded Jacobian. Thus, specifying actions in task space can improve robustness and even accelerate the learning rate by improving sample efficiency [47]. Moreover, learning a policy in task space can ease its transferability to different robots with a varying number of joints, as long as the policy respects the joint boundaries of each robot while working in Cartesian space. Therefore, the action space A is defined as:

$$A = [\Delta x, \Delta y], \tag{3}$$

where Δx and Δy are translational and rotational motions along the Cartesian x and y axes, respectively.

3.1.3. Reward Function

The agent must perform the disassembly of the peg considering the location of the co-worker. Hence, two objectives can be distinguished; one related to the disassembly task itself and the second one that minimizes the robot's collision risk with the human operator. Therefore, the complete reward function can be defined as follows:

$$r = r_{disassembly} + r_{risk}, \tag{4}$$

$r_{disassembly}$ is a continuous reward provided at each timestep t and is defined by the displacement of the peg with respect to the base, d, and penalized by the number of timesteps required to perform the task.

$$r_{disassembly} = 5 \cdot e^{log(d)+4} - w_1 \cdot \frac{timestep}{timestep_{max}}, \tag{5}$$

As shown in Figure 2, the resulting function increases as the extraction is performed in fewer timesteps while it decreases as the timesteps elapse and the robot has not displaced the peg.

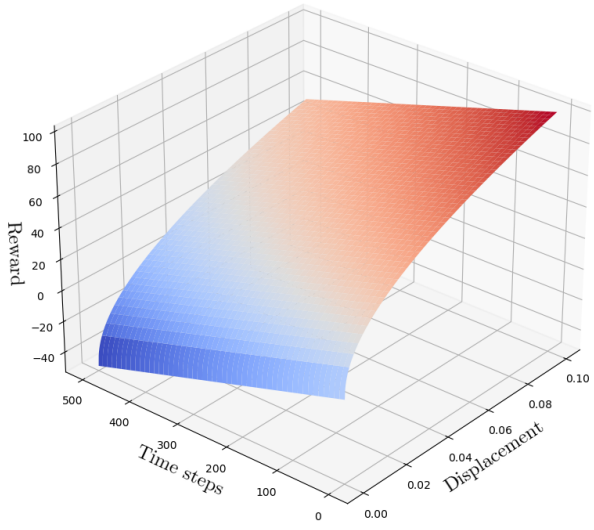

Figure 2. Representation of the disassembly reward function and how it depends on the displacement of the peg with respect to the base and the number of time steps spent for it.

r_{risk} only takes a value if the robot approaches the obstacle. For this purpose, a comparison is made between the previous and current timestep Euclidean distance. This function penalizes the agent according to whether its displacement, d, is greater and, therefore, its Euclidean distance to the obstacle is smaller.

$$r_{risk} = -w_2 \cdot d, \tag{6}$$

In addition, if certain conditions are met, a series of sparse rewards are provided to the agent and the episode is ended.

$$r_{disassembly} = \begin{cases} +200 & \text{if } d > dis_{th} \\ -100 & \text{if } F > F_{th} \\ -100 & \text{if } rot > rot_{th} \\ -100 & \text{if } timestep > timestep_{max} \end{cases}, \tag{7}$$

$$r_{risk} = \{-100 \quad \text{if d}(ee_{x,y}, obs_{x,y}) > safety_{th}\,, \tag{8}$$

where dis_{th} is the disassembly threshold, F is the resultant force applied on the base, F_{th} is the maximum force threshold that can be exerted, rot and rot_{th} are the rotation of the peg in its longitudinal axis at each timestep and the maximum rotation threshold, respectively, $\text{d}(ee_{x,y}, obs_{x,y})$ is the Euclidean distance between the end-effector and the obstacle, and $safety_{th}$ is a set safety distance.

3.2. Reinforcement Learning Agents

Depending on the selected agent, the policy $\pi(a|s)$ may be represented either as a simple function or as a deterministic or stochastic function requiring further computation.

Model-free RL algorithms are classified into policy-based, value-based, and actor-critic methods, where the actor represents the policy, and the critic represents the value function.

Policy-based methods typically work with parameterized functions that directly employ optimization procedures. These functions allow them to generate a continuous spectrum of actions, albeit with high variance. Value-based methods, on the other hand, use temporal difference learning. This type of learning reduces the variance in the estimates of expected returns, although it implies an optimization procedure in each state encountered to find the action that leads to an optimal value. This can be computationally expensive, especially if the task is continuous. Therefore, these methods generally deal with discrete action spaces. Actor-critic algorithms combine the advantages of policy-based and value-based methods. While the parameterized actor estimates continuous actions without the need for optimization over a value function, the critic provides the actor with low-variance knowledge about performance. More precisely, the critic's estimation of expected performance allows the actor to update with lower variance gradients, thus speeding up the learning process.

In this research, three state-of-the-art actor-critic algorithms have been considered: Proximal Policy Optimization (PPO) [48], Deep Deterministic Policy Gradient (DDPG) [49], and Soft Actor-Critic (SAC) [50]. All of them are model-free algorithms. PPO is an on-policy algorithm that tries to compute an update at each step that minimizes the cost function while ensuring that the deviation from the previous policy is relatively small. DDPG and SAC, in turn, are off-policy algorithms, that use a replay buffer memory to store experiences and reuse the most valuable information for efficient training. DDPG is a deterministic algorithm that uses deep function approximators to learn the policy and estimate the value function in continuous, high-dimensional action spaces. Lastly, SAC aims to optimize the maximum entropy. This optimization enhances the exploration and provides the agent with higher generalization capability by visiting states representing extensive learning.

4. Experiments and Results

4.1. Task Description

To prove whether the proposed goal-conditioned RL method, based on state-of-the-art actor-critic algorithms, performs consistently, it was evaluated based on a real use case, namely, the removal of the magnetic gasket attached to refrigerator doors, which allows the doors to close hermetically. This task is currently performed manually. Workers use a screwdriver to pry a corner of the gasket and pull it out (Figure 3a). However, this task can be hazardous and even lead to long-term musculoskeletal injuries, so automating it would improve labor conditions.

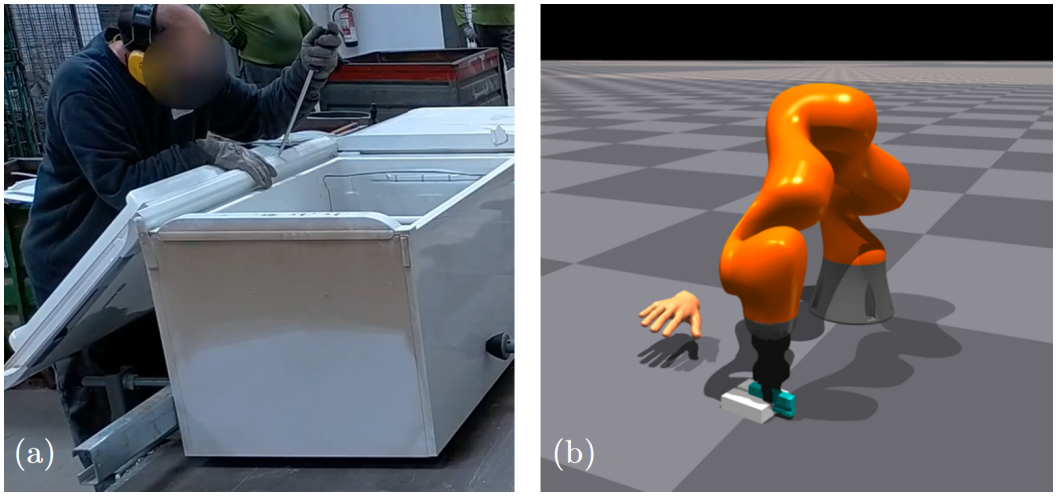

Figure 3. (**a**) Magnetic gasket extraction in an actual disassembly plant. Since the inner magnet of the gaskets cannot be recycled, the gaskets need to be removed from the doors. To do so, the worker uses a screwdriver to pry the magnetic gasket slightly and then pull it out. (**b**) Simplified simulation environment for RL agent learning.

This study sought to perform a proof-of-concept to test the feasibility of RL in contact-rich tasks and collision avoidance simultaneously rather than to solve a real industrial challenge. Therefore, to simplify the complexity of removal, two rigid parts were designed, a fixed slotted base attached to the table and a peg inserted into the slot and gripped by the robot's end-effector. The base and slot dimensions are $0.1 \times 0.1 \times 0.03$ and $0.1 \times 0.02 \times 0.01$ m, respectively. The slot was centered 0.01 m deep with respect to the top face of the base.

The robot should extract the peg towards the opposite side where the co-worker is located, as shown in Figure 3b. For the sake of simplicity, only the worker's hand was modeled, whose position was randomized around a 0.35 m radius circumference after each training episode, and a perturbation was introduced to acquire a more robust policy to deal with image capture noise. This perturbation was a ± 0.05 m random noise in the x and y components of the hand added at each time step to the original position of the obstacle in the episode. Thus, the x and y components of the co-worker's hand could range from 0.3 to 0.4 m at each time step. Similarly, the friction between the base and the peg and their rotation was also randomized, ranging from 0 to 0.05 and rotating up to a maximum of 20°, respectively.

To carry out the training and evaluations, the physical environment of the training process was reproduced using the NVIDIA Isaac Gym (https://developer.nvidia.com/isaac-gym, accessed on 14 September 2022) simulator through the skrl library [51]. All training and evaluations were performed on a workstation with a 3.00 GHz Intel Xeon W-2295 CPU, 126 GB of RAM, and an NVIDIA RTX 6000 GPU with 24 GB VRAM.

4.2. Agent Training

4.2.1. Hyperparameters Selection

Within this case study, a distinction was made between those hyperparameters related to the reward function and those related to the agents employed, namely, PPO, DDPG, and SAC. Tables 1 and 2 show the selected values in both cases.

Table 1. Choice of parameters for reward function.

Parameters	Value
$timestep_{max}$	500
w_1	50
w_2	1000
$disassembly_{th}$	0.1 m
F_{th}	2 N
rot_{th}	7.5°
$safety_{th}$	0.2 m

Table 2. Choice of parameters for agents.

Agent	Parameters	Value
PPO	Memory size	16
	Rollouts	16
	Learning epochs	8
	Discount factor γ	0.99
DDPG	Memory size	50,000
	Batch size	512
	Discount factor γ	0.99
	Learning rate η	10^{-3}
	Noise type	*Ornstein-Uhlenbeck*
	θ	0.15
	σ	0.2
	Base scale	0.1
SAC	Memory size	50,000
	Batch size	512
	Discount factor γ	0.99
	Learning rate η	10^{-3}
	Initial entropy value	0.2

4.2.2. Training

Figure 4 shows the mean and standard deviation of reward received for the three agents in 10 training sessions each. To reduce training times and enrich agent exploration, each training session was run for 1024 environments with different settings, which took about 45 min per session. As can be seen, although PPO is the agent that stabilizes first, eventually all three algorithms converge to a similar value.

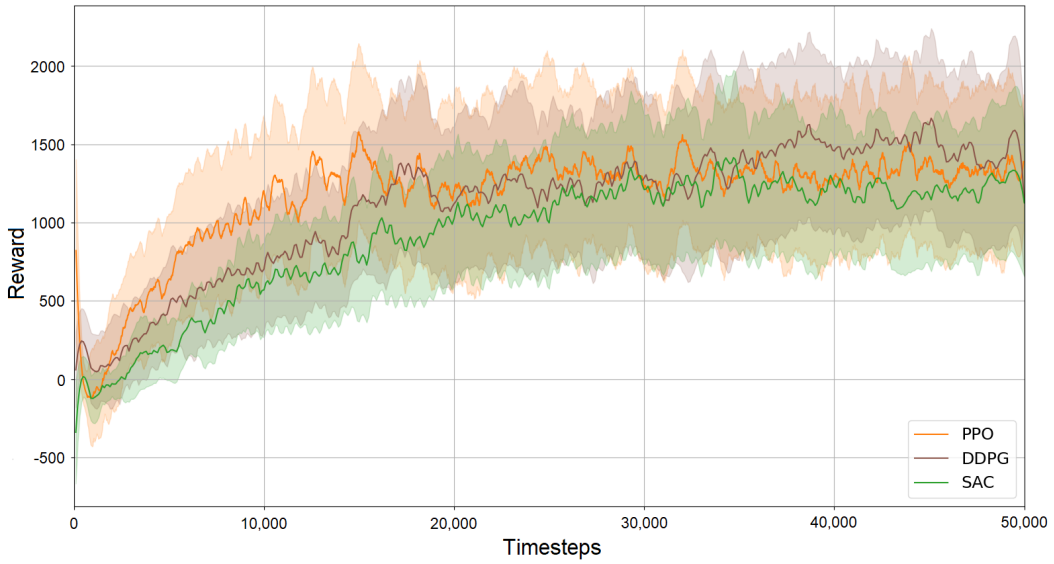

Figure 4. Mean reward and standard deviation of the reward perceived by each of the agents during simulation training.

4.3. Simulation Results

The policies learned were then evaluated with static and dynamic obstacles. Instead of generating dynamic obstacles with a random or linear motion in a single plane, which may underrepresent hazardous situations, actual human movements related to the task to be performed were used. Specifically, three movements were generated through a virtual reality system. The movements envisaged the robot removing the magnetic gasket so that the worker would have to open the refrigerator door, extract the items inside and close the door. For this purpose, two types of refrigerators were considered:

- A refrigerator with two doors, one being the refrigerator and the other the freezer, where the co-worker would open one door, remove drawers and shelves from inside, put them aside, close the door and do the same with the second door.
- A second refrigerator with a single door on which the co-worker could work either to the robot's left or right (Figure 5).

10 evaluations were performed with the best neural weights from each agent training curve. Regarding the evaluations with static obstacles, the three agents executed the task successfully with average success rates of 87.82%, 77.10%, and 84.43% for PPO, DDPG, and SAC, respectively. The violation of the rotation threshold conditioned all those episodes that failed to perform a successful disassembly. Figure 6 shows the position of the obstacle with respect to the rotation of the part in all episodes sampled in a single evaluation for each agent, as well as the box and whiskers chart of the 10 evaluations performed with each agent. These statistics are represented according to the scores obtained by each agent in 30° intervals. The violations occurred mostly when the obstacle was close to the *y*-axis (60–90°, 90–120°, 240–270° and 270–300°), in a neutral standoff position, where the extraction of the part did not pose a risk. This may be due to the uncertainty of the neural networks and the exploration during learning. Nevertheless, the proportion of completed tasks for all the agents could reach a higher value with static, but more realistic positions, placing the obstacle partially to one side of the workpiece. In addition, PPO was the agent with the least variability in scores across the 10 evaluations, with a mean standard deviation of ±3.76. In contrast, SAC and DDPG obtained more distributed values around the means, with mean standard deviations of ±7.74 and ±20.41, respectively.

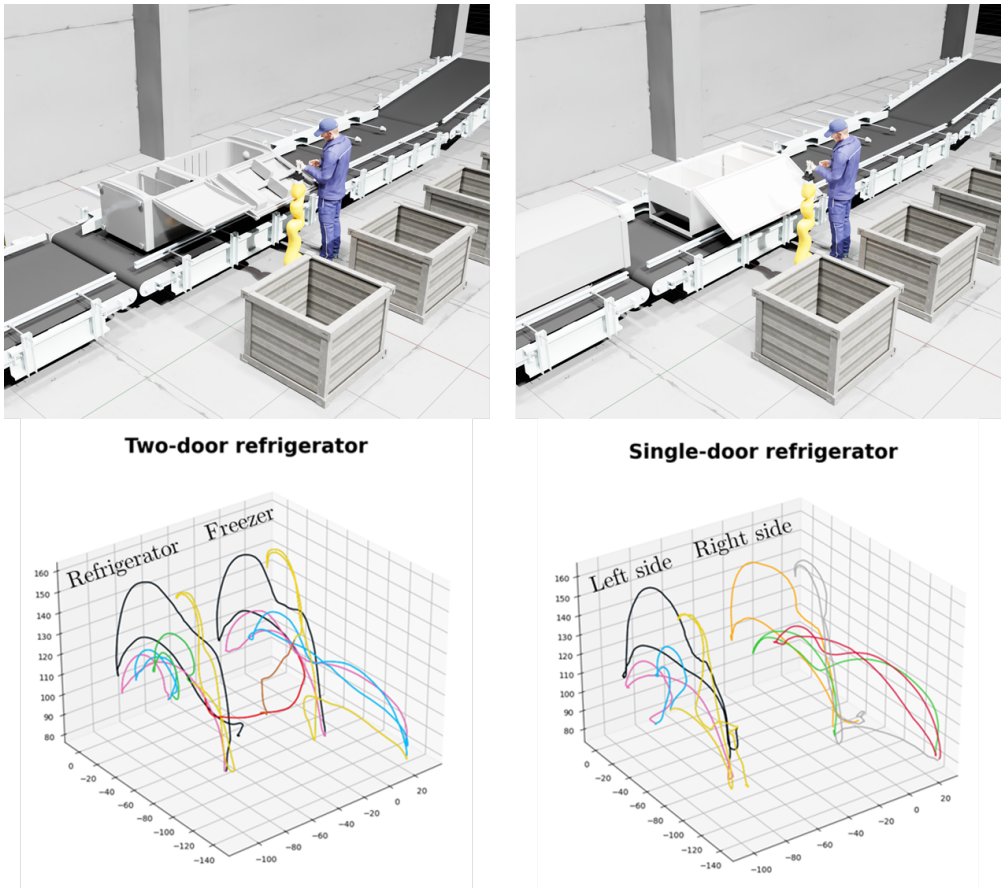

Figure 5. Hand movements acquired from virtual reality system to evaluate the learned policies. In the two-door refrigerator (**bottom left**), the black and yellow lines represent the opening and closing of the doors, respectively, the red line represents the transition of the worker from the refrigerator to the freezer behind the robot, and the pink, blue and green lines show the removal of shelves and drawers inside the refrigerator and freezer. In the single-door refrigerator (**bottom right**), the operator can work either to the robot's left or right. The black and yellow lines on the left again represent the opening and closing of the refrigerator door, respectively, while the pink and blue lines depict the removal of two drawers or shelves. The orange and gray lines on the right also represent the opening and closing of the door, respectively, while the red and green lines indicate the extraction of drawers or shelves.

On the other hand, when it comes to the evaluations with dynamic obstacles, the performance of the three agents dropped significantly, with success rates of 76.31%, 72.64%, and 75.14% for PPO, DDPG, and SAC, respectively. This is primarily due to the design of the reward function and how the continuous movement of the worker's hand causes multiple switches between the disassembly and risk-penalty functions. Nevertheless, the co-worker's safety was almost completely assured during disassembly. In about 97% of the cases in which the episode was considered failed, regardless of the agent, it was the removal of the peg that failed to execute, while the agent was able to direct the removal to the opposite side of the co-worker. In less than 3% of the failed episodes, the extraction was performed correctly, albeit to the co-worker's side.

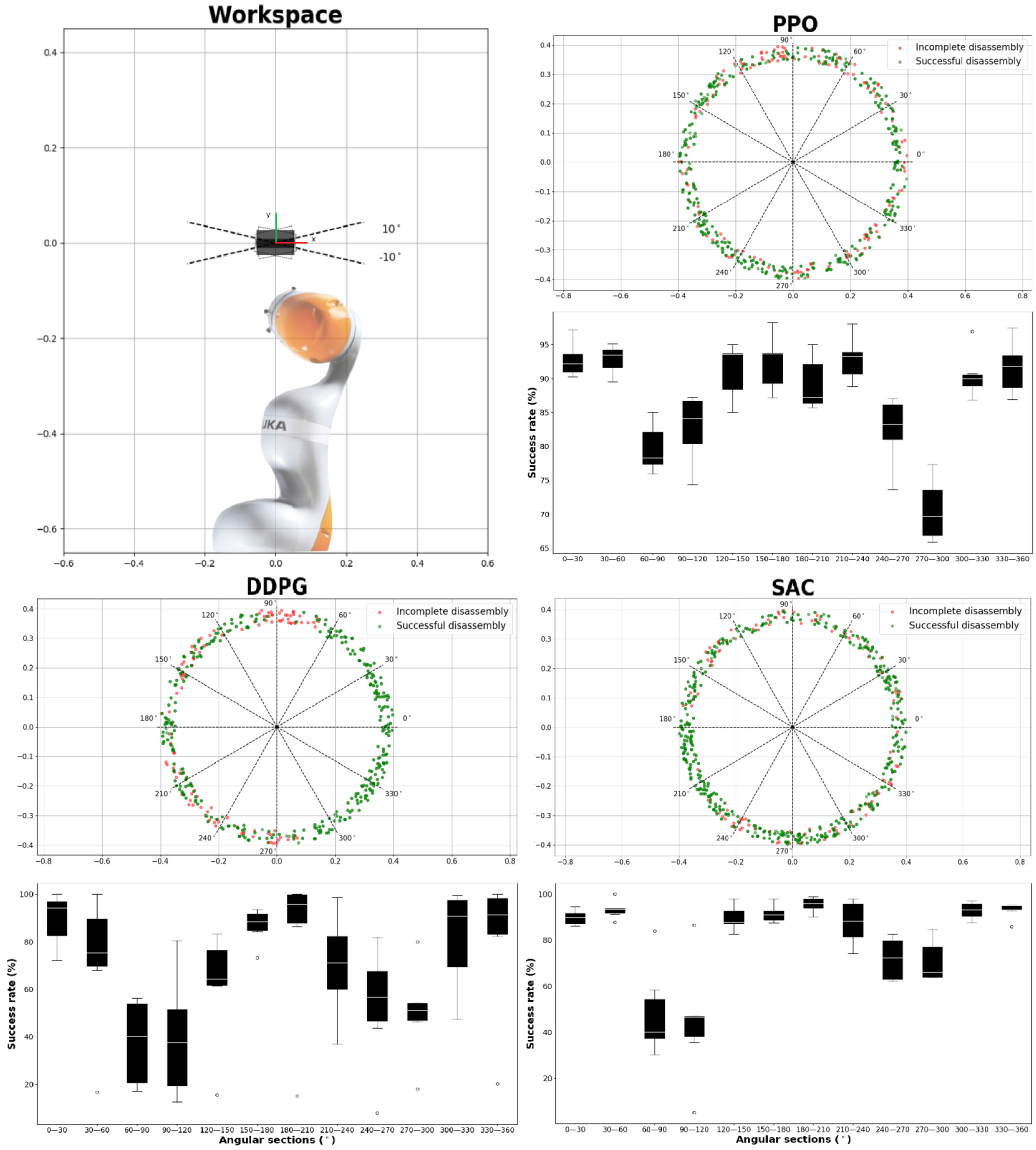

Figure 6. Successful (green points) and incomplete (red points) disassembly trials for each agent in a single evaluation, considering the angle between the co-worker's hand position and the peg's frame at the end of the episode and their respective box and whiskers plots with the averages of the 10 evaluations performed per agent.

4.4. Experiments on the Real System

4.4.1. Simulation to Reality Deployment

ROS is the standard communication infrastructure for software interoperability in robotics and the exchange of data among different applications through a common channel. However, in the field of robot control and ML, there are techniques and/or applications that demand direct or low latency control [52]. This is the case of RL where, at times, the response latency of the robot must be low due to the interaction of an agent with unstructured environments. Due to the communication structure proposed by ROS, its application in RL may be limited by higher response times [53]. Therefore, it is advisable to rely on direct control leading to lower response times in industrial applications where the robot has to act quickly.

Hence, for KUKA LBR Iiwa robot control, a framework was developed that provided a communication interface that not only allowed the development of robotic applications on the robot's own hardware and software, but also allowed external manipulation both via ROS and directly with an Application Programming Interface (API) in Python. For

further information, check the framework's documentation (https://libiiwa.readthedocs.io, accessed on 19 September 2022). For Franka Emika Panda robot control, *frankx* (https://github.com/pantor/frankx, accessed on 19 September 2022) library was employed.

For safe disassembly and as a safety measure against potential collisions with the co-worker, a compliance controller that operated in end-effector position space was set up. When a threshold of F_{max} = 3 N was exceeded, the robot entered a safety stop.

To monitor the co-worker's position in the workspace, a Logitech V-U0028 webcam was placed on the ceiling of the laboratory which, through *openpose* (https://github.com/CMU-Perceptual-Computing-Lab/openpose, accessed on 19 September 2022), tracked the human's motion. As in the simulation training, only the data from the co-worker's right hand was used as part of the observation.

4.4.2. Policy Transferability

The performance of the learned policy was evaluated with different robots (Figure 7). In theory, policies learned in task space can transfer with a higher success rate, abstracting from the kinematics and dynamics of each specific robot model [54]. For this purpose, a zero-shot transfer was performed between the KUKA LBR Iiwa robot, which was used for learning, to the Franka Emika Panda robot. The evaluations were performed with three pairs of disassembly parts, of 10^{-3}, 5×10^{-4} and 2×10^{-4} m tolerance, which affected the friction during the extraction. Specifically, 40 extractions were conducted at every 5° rotation, between $-10°$ and 10° (see Figure 6), of the base and peg. Half of the extractions were carried out with the worker's hand statically positioned near the part and the other half executing a motion similar to that performed on the disassembly plant next to the robot. Static hand evaluations were performed by positioning the hand to the right or left side of the workpiece. All the experiments, in reality, were performed with the neural weights of PPO as it was the agent with the best simulation results. Table 3 gathers the results of the evaluations done with each robot.

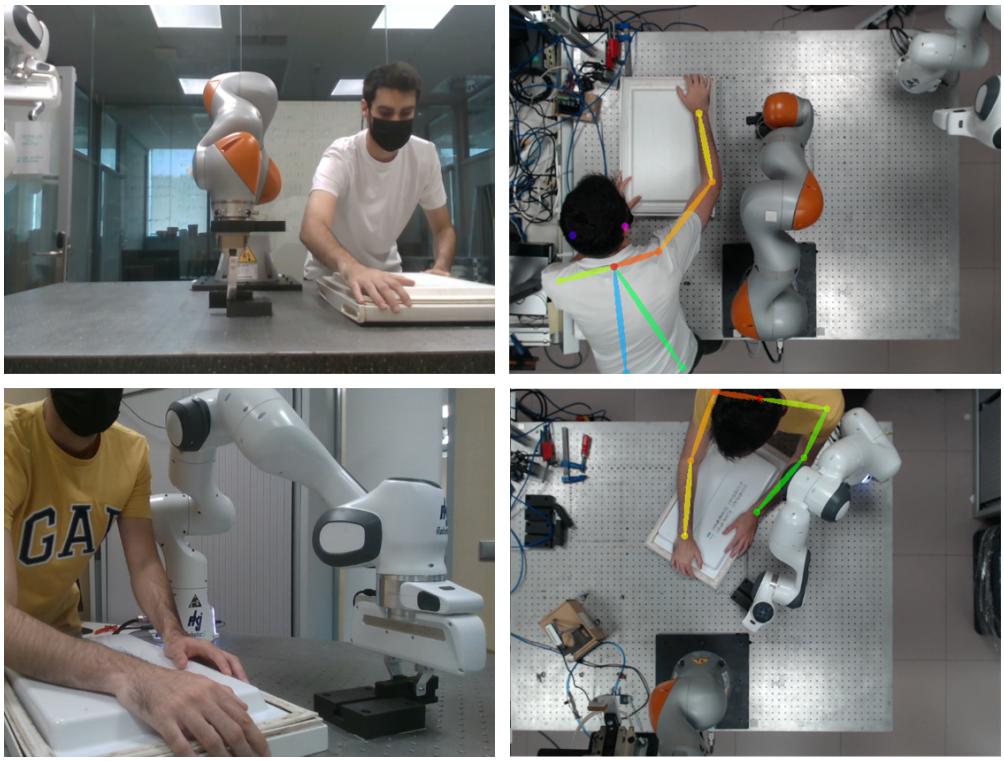

Figure 7. Real-world experiments with two collaborative robots: KUKA LBR Iiwa (**top**) and Franka Emika Panda (**bottom**).

The results showed how the policies learned in the task space can be transferred directly between robots. However, depending on the dimensions of the workspace, the kinematic constraints of each of the robots should be considered.

The KUKA LBR Iiwa robot achieved a disassembly success rate of 95% while avoiding obstacles, 100% with static obstacles, and 90% with dynamic obstacles, when the tolerance between parts was 10^{-3} m. The Franka Emika Panda, in turn, succeeded 93% of the time in disassembly, 94% with static obstacles, and 92% with moving obstacles. In no case was there a collision with the co-worker. The high-efficiency rates of both robots were due to the linear movements executed by both manipulators in reality. It is also worth noting that in evaluating the disassembly with static obstacles in simulation, all failed episodes were caused by the violation of the rotation threshold set as part of the reward. This terminal reward was established since the peg penetrated slightly into the base owing to the simulator's physics, which in reality does not occur.

As the tolerance between parts was reduced, both robots' disassembly success rates dropped significantly. However, the results hardly changed between 5×10^{-4} and 2×10^{-4} tolerances. The KUKA LBR Iiwa achieved success rates of 84.5% and 85% for 5×10^{-4} and 2×10^{-4} tolerances, respectively, while the Franka Emika Panda achieved results of 83% and 82.5%. A plausible explanation for this drop in performance may be related to the fact that the range of actual friction between the two parts due to their tolerances was not completely randomized during simulation learning. Nevertheless, in none of these cases was a collision with the co-worker. Both simulation and experimental results can be found at https://www.youtube.com/watch?v=Sb5sv4PWzhA (accessed on 29 September 2022).

Table 3. Real system evaluations with disassembly parts of different tolerances, and orientations and with both static and dynamic obstacles.

Robot	**10^{-3} m Tolerance Parts Disassembly**									
	−10°		**−5°**		**0°**		**5°**		**10°**	
	S [1]	**D [2]**	**S**	**D**	**S**	**D**	**S**	**D**	**S**	**D**
KUKA LBR Iiwa	20/20 (100%)	17/20 (85%)	20/20 (100%)	19/20 (95%)	20/20 (100%)	19/20 (95%)	20/20 (100%)	18/20 (90%)	20/20 (100%)	17/20 (85%)
Franka Emika Panda	17/20 (85%)	18/20 (90%)	20/20 (100%)	19/20 (95%)	20/20 (100%)	20/20 (100%)	18/20 (90%)	18/20 (90%)	19/20 (95%)	17/20 (85%)
Robot	**5×10^{-4} m Tolerance Parts Disassembly**									
	−10°		**−5°**		**0°**		**5°**		**10°**	
	S	**D**	**S**	**D**	**S**	**D**	**S**	**D**	**S**	**D**
KUKA LBR Iiwa	15/20 (75%)	14/20 (70%)	19/20 (95%)	17/20 (85%)	18/20 (90%)	16/20 (80%)	20/20 (100%)	17/20 (85%)	19/20 (95%)	14/20 (70%)
Franka Emika Panda	14/20 (70%)	14/20 (70%)	19/20 (95%)	17/20 (85%)	18/20 (90%)	17/20 (85%)	19/20 (95%)	16/20 (80%)	17/20 (85%)	15/20 (75%)

Table 3. *Cont.*

Robot	2×10^{-4} m Tolerance Parts Disassembly									
	−10°		−5°		0°		5°		10°	
	S	D	S	D	S	D	S	D	S	D
KUKA LBR Iiwa	15/20 (75%)	15/20 (75%)	20/20 (100%)	17/20 (85%)	19/20 (95%)	14/20 (70%)	20/20 (100%)	17/20 (85%)	20/20 (100%)	13/20 (65%)
Franka Emika Panda	14/20 (70%)	14/20 (70%)	18/20 (90%)	16/20 (80%)	18/20 (90%)	17/20 (85%)	20/20 (100%)	15/20 (75%)	19/20 (95%)	14/20 (70%)

[1] Static obstacle (S). [2] Dynamic obstacle (D).

4.4.3. Baseline Comparison

The proposed approach is an extension of previous research conducted on RL and disassembly [28]. This previous study focused only on disassembly and evaluated the performance of DDPG and Twin Delayed DDPG (TD3) actor-critic algorithms in disassembly parts with different rotations and initial locations both in simulation and in reality. In the experiments, success average disassembly rates of 87.29% and 35.35% were obtained in simulation for DDPG and TD3, respectively, and 94.73% and 36.86% in reality. The improved success rate in performing the task in the real world with respect to the simulation was due to the critical force threshold minor differences used in the environments. Due to the poor performance of TD3, the proposed new approach evaluates and compares DDPG with PPO and SAC, two other agents highly employed in state-of-the-art, for both contact-rich tasks and path planning. In addition, co-worker safety in a shared workspace is included in the agent's reward function, the extraction is generalized to parts with different tolerances and frictions, and the transferability of the policy to two collaborative manipulators is evaluated. Even with all these further considerations, the successful disassembly rate in reality with parts with tolerances of 10^{-3} m using the same robot as in the previous study was 95%, close to the 96% obtained previously, in the disassembly of the same parts.

5. Discussion

The research and application of RL in robotics have exploded over the past few years. However, its potential is still far from reaching, and certain limitations still hinder its use in industrial applications.

The results of this paper show that RL alone can execute contact-rich manipulation tasks while avoiding potential collisions with a co-worker in an HRI scenario and has significant benefits over other current approaches.

A significant benefit is the observed safe performance without the need to model human behavior. Refs. [55,56] addressed controller safety by considering data-driven, such as RL, and model-based control theories. Data-driven approaches attempt to manage uncertainties and reduce the conservatism of the safe controller. Despite the uncertainty introduced by human behavior, the results show that accurate reward shaping and compliant robot behavior that halts its motion in case of collision are sufficient to ensure a safe HRI environment.

Moreover, the proposed approach tries to avoid the activation of safety constraints, such as the safety stops established by the ISO/TS 15066:2016 [57] standard in collaborative robotics in the case, in any of the four collaboration modes defined by the standard, any requirement is violated. By executing the disassembly to the opposite side of the worker, due to the adaptive behavior of the agent to dynamic obstacles, the likelihood of a potential collision is reduced and, therefore, the need to execute a safety stop and stop the workflow.

Another benefit is the agent's ability to generalize. The experiments carried out in reality not only show how the agent can generalize the disassembly to parts with different orientations but also the randomization of the friction between the parts in the learning

process and the introduction of artifacts on the obstacle position measurements, favor the extraction of parts with different tolerances and the achievement of a robust policy in the face of noisy measurements in reality.

The last aspect worth noting is creating the multi-control framework for the direct transfer of actions to the robot exploited by the proposed approach. Since ROS does not satisfy real-time requirements [53], a framework allowing direct control through an API developed in Python was proposed. The low latency in the system's communication infrastructure enables a shorter reaction time for the robot, being able to modify its trajectory to avoid a potential collision in a shorter time range.

A natural concern of the research is related to the complexity of reward shaping. While many authors provide only sparse rewards during learning [35,39], these are not usually successful when learning involves multiple goals. However, this does not necessarily imply the definition of multiple reward sources, each associated with a single goal. Silver et al. [58] suggested that an agent that maximizes reward to achieve its goal might implicitly produce skills that are orthogonal to the agent's main goal and are directed to multiple other pragmatic goals of the agent's intelligence.

On the other hand, to build intelligent collaborative disassembly cells that can be used in real plants, disassembly effectiveness needs to be increased. This will not always be possible, especially when the disassembled parts include flexible elements and time-varying adhesion forces, such as real magnetic gaskets. Therefore, in such situations, the co-worker would have to help the robot to perform the removal. A possible line of research could include the semantics of the co-worker's motion in the observation of the RL agent. By being aware of the human's action, the robot could identify at which point the co-worker also participates in the extraction of the part and let itself be manually guided to finish the disassembly in case it cannot do it by itself.

6. Conclusions

This paper investigates the applicability of deep RL in simultaneous disassembly and collision avoidance tasks in an HRI workspace. For this purpose, the performance of three state-of-the-art model-free agents with both static and dynamic obstacles constrained by the task to be performed by the co-worker is evaluated in simulation and reality with different collaborative robots.

Although the results show a promising avenue of research, and even the proposed approach outperforms the previous study on disassembly, this paper has certain limitations. Among them, the most noteworthy is simplifying the actual use case considered. Although this study aims to analyze the feasibility of the RL as a tool to simultaneously perform contact-rich manipulation and collision avoidance, the control policy has been evaluated in an environment far from reality. On the other hand, the robot is currently agnostic to the task undertaken by the co-worker. Having richer observation input could lead to more intelligent agents whose decision-making would not be based solely on collision avoidance. Therefore, as future lines of research, it would be desirable to train the robot to perform the disassembly of real magnetic gaskets or even other contact-rich tasks and to be aware of its environment. This latter line of research may involve the robot identifying the task being performed by the co-worker at all times and letting itself be assisted in case the manipulator is unable to perform the disassembly by itself, resulting in a higher success rate in gasket removal. This approach would also entail employing skeleton tracking to identify the human operator's gestures. This information could also be used to increase the safety of the system by considering the entire body of the co-worker. Lastly, considering the GPU's power consumption, it is suggested to explore the use of neuromorphic computing, which can be more energy-efficient [59].

Supplementary Materials: The following are available online at https://www.mdpi.com/article/10.3390/app122211610/s1, Video S1 (available online at https://www.youtube.com/watch?v=Sb5sv4PWzhA, (accessed on 29 September 2022)).

Author Contributions: Conceptualization, Í.E.-A. and N.A.-A.; investigation, Í.E.-A.; methodology, Í.E.-A. and A.S.-M.; software, Í.E.-A. and A.S.-M.; supervision, D.C., S.B. and N.A.-A.; visualization, Í.E.-A. and I.I.-H.; writing—original draft preparation, Í.E.-A.; writing—review and editing, Í.E.-A., D.C., S.B. and N.A.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the H2020-WIDESPREAD project no. 857061 "Networking for Research and Development of Human Interactive and Sensitive Robotics Taking Advantage of Additive Manufacturing-R2P2", the H2020-ECSEL JU project no. 876852 "Verification and Validation of Automated Systems' Safety and Security-VALU3S", and Basque Government Department of Economic Development, Sustainability, and Environment through the Bikaintek 2020 program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
APF	Artificial Potential Field
DDPG	Deep Deterministic Policy Gradient
EoL	End-of-Life
HRI	Human-Robot Interaction
MDP	Markov Decision Process
ML	Machine Learning
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
ROS	Robot Operating System
RRT	Rapid-exploring Random Trees
SAC	Soft Actor-Critic
TD3	Twin Delayed DDPG
WEEE	Waste Electrical and Electronic Equipment

References

1. Waste from Electrical and Electronic Equipment (WEEE). Available online: https://ec.europa.eu/environment/topics/waste-and-recycling/waste-electrical-and-electronic-equipment-weee_en (accessed on 7 August 2022).
2. Global Forum Tokyo Issues Paper 30-5-2014.pdf. Available online: https://www.oecd.org/environment/waste/Global%20Forum%20Tokyo%20Issues%20Paper%2030-5-2014.pdf (accessed on 7 August 2022).
3. Vongbunyong, S.; Chen, W.H. Disassembly automation. In *Disassembly Automation*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 25–54.
4. Hjorth, S.; Chrysostomou, D. Human–robot collaboration in industrial environments: A literature review on non-destructive disassembly. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102208. [CrossRef]
5. Shailaja, K.; Seetharamulu, B.; Jabbar, M. Machine learning in healthcare: A review. In Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 29–31 March 2018; pp. 910–914.
6. Bachute, M.R.; Subhedar, J.M. Autonomous driving architectures: Insights of machine learning and deep learning algorithms. *Mach. Learn. Appl.* **2021**, *6*, 100164. [CrossRef]
7. Wang, W.; Siau, K. Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: A review and research agenda. *J. Database Manag.* **2019**, *30*, 61–79. [CrossRef]
8. Jurgenson, T.; Avner, O.; Groshev, E.; Tamar, A. Sub-Goal Trees a Framework for Goal-Based Reinforcement Learning. In Proceedings of the International Conference on Machine Learning, Online, 13–18 July 2020; pp. 5020–5030.
9. Yang, X., Ji, Z., Wu, J., Lai, Y.-K., Wei, C., Liu, G., Setchi, R. Hierarchical Reinforcement Learning With Universal Policies for Multistep Robotic Manipulation. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 4727–4741. [CrossRef] [PubMed]
10. Liu, M.; Zhu, M.; Zhang, W. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv* **2022**, arxiv:2201.08299.
11. Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Pieter Abbeel, O.; Zaremba, W. Hindsight experience replay. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.

12. Zhang, H.; Wang, F.; Wang, J.; Cui, B. Robot grasping method optimization using improved deep deterministic policy gradient algorithm of deep reinforcement learning. *Rev. Sci. Instrum.* **2021**, *92*, 025114. [CrossRef]
13. Vulin, N.; Christen, S.; Stevšić, S.; Hilliges, O. Improved learning of robot manipulation tasks via tactile intrinsic motivation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2194–2201. [CrossRef]
14. Kim, Y.L.; Ahn, K.H.; Song, J.B. Reinforcement learning based on movement primitives for contact tasks. *Robot.-Comput.-Integr. Manuf.* **2020**, *62*, 101863. [CrossRef]
15. Luo, J.; Solowjow, E.; Wen, C.; Ojea, J.A.; Agogino, A.M.; Tamar, A.; Abbeel, P. Reinforcement learning on variable impedance controller for high-precision robotic assembly. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QB, Canada, 20–25 May 2019; pp. 3080–3087.
16. Li, X.; Xiao, J.; Zhao, W.; Liu, H.; Wang, G. Multiple peg-in-hole compliant assembly based on a learning-accelerated deep deterministic policy gradient strategy. *Ind. Robot. Int. J. Robot. Res. Appl.* **2021**, *49*, 54–64. [CrossRef]
17. Ennen, P.; Bresenitz, P.; Vossen, R.; Hees, F. Learning robust manipulation skills with guided policy search via generative motor reflexes. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QB, Canda, 20–25 May 2019; pp. 7851–7857.
18. Fan, Y.; Luo, J.; Tomizuka, M. A learning framework for high precision industrial assembly. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QB, Canda, 20–25 May 2019; pp. 811–817.
19. Khader, S.A.; Yin, H.; Falco, P.; Kragic, D. Stability-guaranteed reinforcement learning for contact-rich manipulation. *IEEE Robot. Autom. Lett.* **2020**, *6*, 1–8. [CrossRef]
20. Khader, S.A.; Yin, H.; Falco, P.; Kragic, D. Learning deep energy shaping policies for stability-guaranteed manipulation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 8583–8590. [CrossRef]
21. Ren, T.; Dong, Y.; Wu, D.; Chen, K. Learning-based variable compliance control for robotic assembly. *J. Mech. Robot.* **2018**, *10*, 061008. [CrossRef]
22. Wang, Y.; Beltran-Hernandez, C.C.; Wan, W.; Harada, K. Hybrid Trajectory and Force Learning of Complex Assembly Tasks: A Combined Learning Framework. *IEEE Access* **2021**, *9*, 60175–60186. [CrossRef]
23. Zhao, T.Z.; Luo, J.; Sushkov, O.; Pevceviciute, R.; Heess, N.; Scholz, J.; Schaal, S.; Levine, S. Offline meta-reinforcement learning for industrial insertion. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022, 6386–6393.
24. Beltran-Hernandez, C.C.; Petit, D.; Ramirez-Alpizar, I.G.; Harada, K. Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach. *Appl. Sci.* **2020**, *10*, 6923. [CrossRef]
25. Kristensen, C.B.; Sørensen, F.A.; Nielsen, H.B.; Andersen, M.S.; Bendtsen, S.P.; Bøgh, S. Towards a robot simulation framework for e-waste disassembly using reinforcement learning. *Procedia Manuf.* **2019**, *38*, 225–232. [CrossRef]
26. Simonič, M.; Žlajpah, L.; Ude, A.; Nemec, B. Autonomous Learning of Assembly Tasks from the Corresponding Disassembly Tasks. In Proceedings of the 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), Toronto, ON, Canada, 15–17 October 2019; pp. 230–236.
27. Herold, R.; Wang, Y.; Pham, D.; Huang, J.; Ji, C.; Su, S. Using active adjustment and compliance in robotic disassembly. In *Industry 4.0–Shaping The Future of The Digital World*; CRC Press: Boca Raton, FL, USA, 2020; pp. 101–105. .
28. Serrano-Muñoz, A.; Arana-Arexolaleiba, N.; Chrysostomou, D.; Bøgh, S. Learning and generalising object extraction skill for contact-rich disassembly tasks: An introductory study. *Int. J. Adv. Manuf. Technol.* **2021**, 1–13. [CrossRef]
29. Bonilla, M.; Pallottino, L.; Bicchi, A. Noninteracting constrained motion planning and control for robot manipulators. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4038–4043.
30. Lin, H.C.; Liu, C.; Fan, Y.; Tomizuka, M. Real-time collision avoidance algorithm on industrial manipulators. In Proceedings of the 2017 IEEE Conference on Control Technology and Applications (CCTA), Hawaii, HI, USA, 27–30 August 2017; pp. 1294–1299.
31. Chen, J.H.; Song, K.T. Collision-free motion planning for human-robot collaborative safety under cartesian constraint. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 4348–4354.
32. Yasar, M.S.; Iqbal, T. A scalable approach to predict multi-agent motion for human-robot collaboration. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1686–1693. [CrossRef]
33. Buerkle, A.; Eaton, W.; Lohse, N.; Bamber, T.; Ferreira, P. EEG based arm movement intention recognition towards enhanced safety in symbiotic Human-Robot Collaboration. *Robot.-Comput.-Integr. Manuf.* **2021**, *70*, 102137. [CrossRef]
34. Li, Q.; Nie, J.; Wang, H.; Lu, X.; Song, S. Manipulator Motion Planning based on Actor-Critic Reinforcement Learning. In Proceedings of the 2021 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021; pp. 4248–4254.
35. Prianto, E., Kim, M., Park, J.-H., Bae, J.-H., Kim, J.-S. Path planning for multi-arm manipulators using deep reinforcement learning: Soft actor–critic with hindsight experience replay. *Sensors* **2020**, *20*, 5911. [CrossRef]
36. Zhou, D.; Jia, R.; Yao, H.; Xie, M. Robotic Arm Motion Planning Based on Residual Reinforcement Learning. In Proceedings of the 2021 13th International Conference on Computer and Automation Engineering (ICCAE), Melbourne, Australia, 20–22 March 2021; pp. 89–94.
37. Zhou, D.; Jia, R.; Yao, H. Robotic Arm Motion Planning Based on Curriculum Reinforcement Learning. In Proceedings of the 2021 6th International Conference on Control and Robotics Engineering (ICCRE), Beijing, China, 16–18 April 2021; pp. 44–49.

38. El-Shamouty, M.; Wu, X.; Yang, S.; Albus, M.; Huber, M.F. Towards safe human-robot collaboration using deep reinforcement learning. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 30–31 August 2020; pp. 4899–4905.
39. Prianto, E.; Park, J.-H.; Bae, J.-H.; Kim, J.-S. Deep Reinforcement Learning-Based Path Planning for Multi-Arm Manipulators with Periodically Moving Obstacles. *Appl. Sci.* **2021**, *11*, 2587. [CrossRef]
40. Sangiovanni, B.; Rendiniello, A.; Incremona, G.P.; Ferrara, A.; Piastra, M. Deep reinforcement learning for collision avoidance of robotic manipulators. In Proceedings of the 2018 European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018; pp. 2063–2068.
41. Sangiovanni, B.; Incremona, G.P.; Piastra, M.; Ferrara, A. Self-configuring robot path planning with obstacle avoidance via deep reinforcement learning. *IEEE Control. Syst. Lett.* **2020**, *5*, 397–402. [CrossRef]
42. Xiong, B.; Liu, Q.; Xu, W.; Yao, B.; Liu, Z.; Zhou, Z. Deep reinforcement learning-based safe interaction for industrial human-robot collaboration. In Proceedings of the 49th International Conference on Computers and Industrial Engineering, Beijing, China, 18–21 October 2019; Volume 49, pp. 1–13.
43. Zhao, X.; Fan, T.; Li, Y.; Zheng, Y.; Pan, J. An Efficient and Responsive Robot Motion Controller for Safe Human-Robot Collaboration. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6068–6075. [CrossRef]
44. Yamada, J.; Lee, Y.; Salhotra, G.; Pertsch, K.; Pflueger, M.; Sukhatme, G.S.; Lim, J.J.; Englert, P. Motion planner augmented reinforcement learning for robot manipulation in obstructed environments. *arXiv* **2020**, arXiv:2010.11940.
45. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; Cambridge University Press: Cambridge, MA, USA, 2018; p. 1.
46. Thomas, G.; Chien, M.; Tamar, A.; Ojea, J.A.; Abbeel, P. Learning robotic assembly from cad. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3524–3531.
47. Spector, O.; Zacksenhouse, M. Deep reinforcement learning for contact-rich skills using compliant movement primitives. *arXiv* **2020**, arXiv:2008.13223.s.
48. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
49. Lillicrap, T. P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
50. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.
51. Serrano-Muñoz, A.; Arana-Arexolaleiba, N.; Chrysostomou, D.; Bøgh, S. skrl: Modular and Flexible Library for Reinforcement Learning. *arXiv* **2022**, arXiv:2202.03825.
52. Baklouti, S.; Gallot, G.; Viaud, J.; Subrin, K. On the Improvement of Ros-Based Control for Teleoperated Yaskawa Robots. *Appl. Sci.* **2021**, *11*, 7190. [CrossRef]
53. Park, J.; Delgado, R.; Choi, B.W. Real-time characteristics of ROS 2.0 in multiagent robot systems: an empirical study. *IEEE Access* **2020**, *8*, 154637–154651. [CrossRef]
54. Martín-Martín, R.; Lee, M.A.; Gardner, R.; Savarese, S.; Bohg, J.; Garg, A. Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 1010–1017.
55. Garcıa, J., Fernández, F. A Comprehensive Survey on Safe Reinforcement Learning. *J. Mach. Learn. Res.* **2015**, *16*, 1437–1480.
56. Brunke, L.; Greeff, M.; Hall, A.W.; Yuan, Z.; Zhou, S.; Panerati, J.; Schoellig, A.P. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. *arXiv* **2022**, arxiv:abs/2108.06266.
57. *ISO/TS 15066*; Robots and Robotic Devices—Collaborative Robots. International Organization for Standardization: Geneva, Switzerland, 2016.
58. Silver, D.; Singh, S.; Precup, D.; Sutton, R.S. Reward is enough. *Artif. Intell.* **2021**, *299*, 103535. [CrossRef]
59. Rao, A.; Plank, P.; Wild, A.; Mass, W. A Long Short-Term Memory for AI Applications in Spike-based Neuromorphic Hardware. *Nat. Mach. Intell.* **2022**, *4*, 467–479. [CrossRef]