frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 23 March 2022
doi: 10.3389/frobt.2022.830007

Learning to Centralize Dual-Arm Assembly

Marvin Alles and Elie Aljalbout*

Technical University of Munich, Munich, Germany

Robotic manipulators are widely used in modern manufacturing processes. However, their deployment in unstructured environments remains an open problem. To deal with the variety, complexity, and uncertainty of real-world manipulation tasks, it is essential to develop a flexible framework with reduced assumptions on the environment characteristics. In recent years, reinforcement learning (RL) has shown great results for single-arm robotic manipulation. However, research focusing on dual-arm manipulation is still rare. From a classical control perspective, solving such tasks often involves complex modeling of interactions between two manipulators and the objects encountered in the tasks, as well as the two robots coupling at a control level. Instead, in this work, we explore the applicability of model-free RL to dual-arm assembly. As we aim to contribute toward an approach that is not limited to dual-arm assembly but dual-arm manipulation in general, we keep modeling efforts at a minimum. Hence, to avoid modeling the interaction between the two robots and the used assembly tools, we present a modular approach with two decentralized single-arm controllers, which are coupled using a single centralized learned policy. We reduce modeling effort to a minimum by using sparse rewards only. Our architecture enables successful assembly and simple transfer from simulation to the real world. We demonstrate the effectiveness of the framework on dual-arm peg-in-hole and analyze sample efficiency and success rates for different action spaces. Moreover, we compare results on different clearances and showcase disturbance recovery and robustness when dealing with position uncertainties. Finally, we zero-shot transfer policies trained in simulation to the real world and evaluate their performance. Videos of the experiments are available at the project website (https://sites.google.com/view/dual-arm-assembly/home).

Keywords: dual-arm manipulation, reinforcement learning, sim-to-real transfer, robotics, assembly, centralized control, peg-in-hole

OPEN ACCESS

Edited by:
Tadahiro Taniguchi, Ritsumeikan University, Japan

Reviewed by:
Kazutoshi Tanaka, OMRON SINIC X Corporation, Japan
Gustavo Alfonso Garcia Ricardez, Ritsumeikan University, Japan

****Correspondence:***
Marvin Alles
marvin.alles@tum.de

Specialty section:
This article was submitted to Robot Learning and Evolution, a section of the journal Frontiers in Robotics and AI

Received: *06 December 2021*
Accepted: *11 February 2022*
Published: *23 March 2022*

Citation:
Alles M and Aljalbout E (2022) Learning to Centralize Dual-Arm Assembly. Front. Robot. AI 9:830007. doi: 10.3389/frobt.2022.830007

1 INTRODUCTION

In recent years, robotic manipulation has been an active field of research. However, work focusing on dual-arm manipulation is still rare and limited. A second robotic arm enhances dexterity but also introduces new challenges and extra modeling efforts, such as additional degrees of freedom and interactions between manipulators. Thus, it is common to use a complex task-specific control structure with multiple control loops (Suomalainen et al., 2019; Zhang et al., 2017; Park et al., 2014). However, such methods are usually restricted to certain classes of tasks that often assume access to accurate models of the objects involved in the task and the interaction dynamics between the two robots (Pairet et al., 2019; Caccavale and Villani, 2001; Caccavale

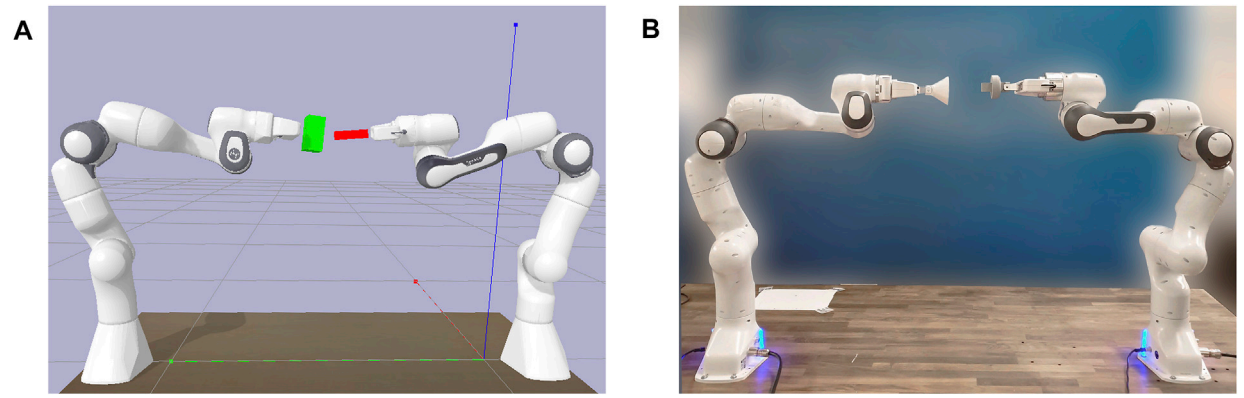

FIGURE 1 | Simulation-to-real transfer. The policy is trained in simulation **(A)** to perform dual-arm peg-in-hole and transferred to the real world **(B)** without additional training.

et al., 2008; Erhart et al., 2013; Heck et al., 2013; Ren et al., 2015; Bjerkeng et al., 2014). In this work, we focus on task-agnostic methods for dual-arm manipulation. Hence, throughout this work, we restrict task-related modeling to a minimum. As real-world robot learning experiments could be very expensive, we only experiment with dual-arm assembly, but restrict ourselves from including or modeling any kind of knowledge specific to this task.

To this end, deep reinforcement learning (RL) is a promising approach to tackle this problem. Thereby, manipulation tasks can be learned from scratch by interaction with the environment. However, deep RL alone would require a lot of training samples, which are expensive to collect in a real-world setup (Zhu et al. 2020, Dulac-Arnold et al. 2020, Ibarz et al. 2021). Instead, it is preferable to introduce inductive biases into our architecture as to facilitate the learning process. Namely, we train a policy network only to generate high-level trajectories and use well-established control techniques to track those trajectories. Such a modular architecture also allows for zero-shot sim-to-real transfer. This enables us to do all the training in simulation.

In general, we distinguish between decentralization and centralization on both the control level and policy level. On a control level, centralized control requires large modeling efforts and is not task agnostic (Caccavale and Villani, 2001; Caccavale et al., 2008; Erhart et al., 2013; Heck et al., 2013; Bjerkeng et al., 2014; Ren et al., 2015), hence our work considers a decentralized approach. With that in mind, two general paradigms are conceivable: the first one involves two separate decoupled RL agents that can be trained in a multi-agent RL setting, and the second one utilizes a single policy controlling both arms. The latter is more feasible as it couples control of both manipulators through a policy network, resulting in an overall centralized method, and thus increases precision and efficiency. Our method is based on the latter approach and attempts to learn a single policy using off-policy RL. Intuitively, such an approach can be thought of as a way to centralize decentralized control based on RL.

This paper aims at exploring the applicability of deep RL to dual-arm assembly. Hence, we propose a framework to learn a policy for this task based on a combination of recent advances in RL and well-established control techniques. To reduce the need for task-specific knowledge and to avoid introducing additional bias in the learning process, we test our methods solely with sparse rewards. Nevertheless, only receiving a reward after successfully solving the task provides less guidance to the agent as no intermediate feedback is given. Thus, the already challenging task of dual-arm manipulation becomes more complicated and sample-inefficient. That is, why we rely on simulation to train our policy and transfer the results to the real-world (**Figure 1**). Moreover, we design our framework with the goal of avoiding elaborate sim-to-real transfer procedures.

To demonstrate the effectiveness of our approach we evaluate our method on a dual-arm peg-in-hole task, as it requires high dexterity to manipulate two objects with small clearances under consideration of contact forces. We first use PyBullet (Coumans and Bai, 2016) to create a real-time physics simulation environment and analyze the proposed approach with a focus on sample efficiency, performance and robustness. We then test the learned behavior in a real-world setup with two Franka Emika Panda robots and demonstrate the feasibility of our method under minimized sim-to-real transfer efforts. Our contributions can be summarized as follows:

- We explore and formulate a new paradigm for learning dual-arm assembly tasks.
- We compare the performance of different action spaces and controllers on the success and robustness of the learned policies.
- We show that it is possible to zero-shot transfer policies trained in simulation to the real-world, when using a suitable and abstract action space.
- To our knowledge, our work is the first to explore the applicability of model-free RL to contact-rich dual-arm manipulation tasks.

2 RELATED WORK

Dual-arm manipulation is a challenging area of research, which can be divided into decentralized and centralized approaches. The first one utilizes independent controllers for each robot with explicit (Petitti et al., 2016) or implicit (Wang and Schwager, 2014; Tsiamis et al., 2015) communication channels and is often combined with leader/follower behavior (Suomalainen et al., 2019; Wang and Schwager, 2014). Despite the resulting improvements in scalability and variability, decentralized control hardly reaches the efficiency and precision of centralized control, which integrates the control of both manipulators in a central unit. Among feasible manipulation objectives, peg-in-hole insertion can be seen as a benchmark since it requires accurate positioning, grasping, and handling of objects in contact-rich situations. Therefore, we select the task of dual-arm peg-in-hole to evaluate the performance of our approach.

2.1 Single-Arm Peg-in-Hole

As research focusing on dual-arm peg-in-hole assembly is rare and mostly limited to extensive modeling (Suomalainen et al., 2019; Park et al., 2014; Zhang et al., 2017), research on classical peg-in-hole assembly with a single robotic arm provides a perspective on model-free approaches based on reinforcement learning. Vecerík et al. (2017) and Schoettler et al. (2019) show that sparse rewards are sufficient to successfully learn a policy for an insertion task if combined with learning from demonstrations. The work in Schoettler et al. (2019) uses residual reinforcement learning to leverage classical control, which performs well given sparse rewards only and provides a hint that the choice of action space can be crucial. An evaluation of action spaces on the task of single-arm peg-in-hole with a clearance of 2 mm and a shaped reward function is presented in Varin et al. (2019), where Cartesian impedance control performs best. Moreover, Beltran-Hernandez et al. (2020) apply position-force control with model-free reinforcement learning for peg-in-hole with a focus on transfer-learning and domain randomization.

2.2 Decentralized Dual-Arm Manipulation

In the work by Suomalainen et al. (2019), a decentralized approach for the dual-arm peg-in-hole task is proposed. The method is based on a leader/follower architecture. Hence, no explicit coupling between both manipulators is required. The leader would perform the insertion, and the follower would hold its position and be compliant with the applied forces. Similar to the previously mentioned work, Zhang et al. (2017) utilizes a decentralized approach, where the hole keeps the desired position with low compliance and the peg is steered in a spiral-screw motion toward insertion with high compliance. However, despite reducing the necessity to model their interaction, both approaches lack dexterity, i.e., there is only one robot actively acting in the environment. In a general pipeline, there should be enough flexibility for both arms to be actively contributing toward the objective. Furthermore, Park et al. (2014) present a method based on decomposing the task into phases and utilizing a sophisticated control flow for the whole assembly process. Despite reducing efforts in modeling the interaction explicitly, the control flow is only engineered for one specific task and lacks dexterity as movements are bound to the preprogrammed procedure.

2.3 Centralized Dual-Arm Manipulation

Work on centralized dual-arm manipulation focuses on cooperative single object manipulation; hence, the applicability is limited to a few use-cases. Pairet et al. (2019) propose a learning-based approach and evaluate their method on a synthetic pick-and-place task. A set of primitive behaviors are demonstrated to the robot by a human; the robot combines those behaviors and tries to solve a given task. Finally, an evaluator measures its performance and decides if further demonstrations are necessary. The approach has promising potential toward more robust and less task-specific dual-arm manipulation. However, besides the required modeling efforts, it is limited by the human teaching process, which introduces an additional set of assumptions, limiting its applicability to semi-structured environments. Besides that, classical methods for cooperative single object manipulation with centralized control highly rely on accurate and complex modeling of the underlying system dynamics (Caccavale and Villani, 2001; Caccavale et al., 2008; Erhart et al., 2013; Heck et al., 2013; Bjerkeng et al., 2014; Ren et al., 2015).

2.4 Sim-to-Real Transfer

Sample inefficiency is one of the main challenges of deep RL algorithms. The problem is even worse for robotic tasks, which involve high-dimensional states and actions as well as complex dynamics. This motivates the use of simulation for data collection and training. However, due to the inaccuracies in the physics modeling and image rendering in simulation, policies trained in simulation tend to fail in the real world. This is usually referred to as the "reality gap." The most popular paradigm to approach this problem is domain randomization (Tobin et al., 2017). The main goal of domain randomization is to subject the agent to samples based on diverse simulation parameters concerning the object (Tobin et al., 2017) and the dynamics properties (Peng et al., 2018). By doing so, the learned policy is supposed to be able to generalize to the different physical properties of real-world tasks. Recent work has explored active parameter sampling strategies as to dedicate more training time for troublesome parameter settings (Mehta et al., 2020). Another approach for sim-to-real transfer is system modularity (Clavera et al., 2017). Here, a policy is split into different modules responsible for different objectives such as pose detection, online motion planning, and control. Only components that will not suffer from the reality gap are trained in simulation and the rest is adapted or tailor-made for the real-world setup. This comes in contrast to the most common end-to-end training in deep RL (Levine et al., 2016). In our work, we use a modular architecture to enable zero-shot sim-to-real transfer. Namely, we parameterize the controllers differently in simulation compared with the real world to allow using the same high-level policy network.

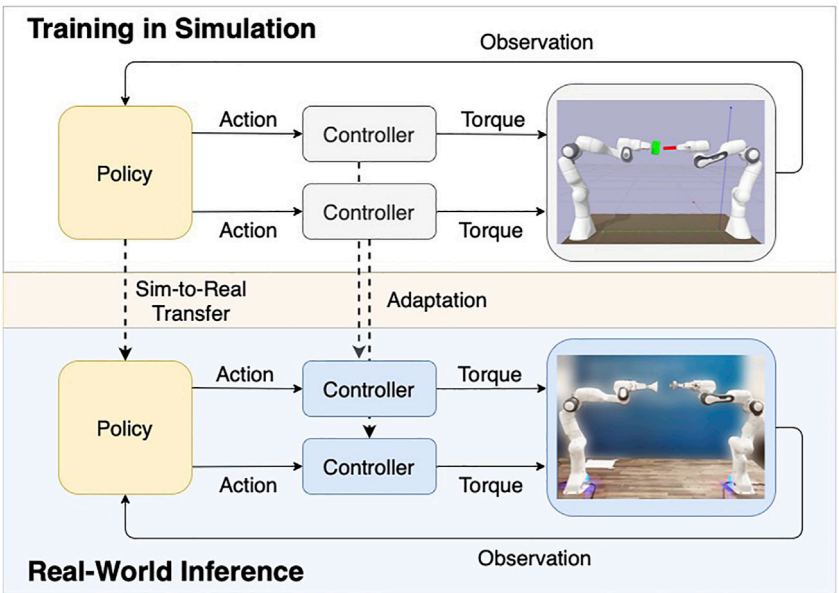

FIGURE 2 | The diagram gives an overview of our method by showing the interaction between policy, controllers, robots, and the environments (simulation and real world). Our framework uses two decentralized single-arm controllers to avoid modeling the task-specific system dynamics and relies on a centralized policy to account for the overall interaction. Based on this architecture, we train our policy in simulation and zero-shot transfer it to the real-world environment. The sim-to-real transfer requires only minor adaptations of the controller parameters, as illustrated by the dashed line.

2.5 Hierarchical Reinforcement Learning

Hierarchical reinforcement learning (HRL) is very frequently used in robotics (Beyret et al., 2019; Bischoff et al., 2013; Jain et al., 2019). These methods typically introduce policies at different layers of abstractions and different time resolutions to break down the complexity of the overall learned behavior. Model-free HRL approaches can either attempt to formulate joint value functions over all the policies such as in Dietterich (2000) or rely on algorithms for hierarchical policy gradients, such as Ghavamzadeh and Mahadevan (2003). Furthermore, Parr and Russell (1998) illustrate how the modularity of HRL methods could enable transferring learned knowledge through component recombination. End et al. (2017) propose a method for autonomously discovering diverse sets of subpolicies and their activation policies. In this work, we design a method with two levels of abstractions. The first one is a learned model-free policy outputting high-level control commands/targets that are carried out by the low-level policy, which is a well-defined controller. Note that the policy used at the second layer of abstraction is not learned but instead designed based on well-established control methods. This enables sample-efficient learning and simple sim-to-real transfer.

Despite various contributions toward a general framework for dual-arm manipulation, we do not know of any work that is task agnostic, does not require explicit modeling of interaction dynamics, and is centralized. Therefore, this work aims at proposing a unified pipeline for dual-arm manipulation based on a minimal set of assumptions. To the best of our knowledge, no prior work exists on contact-rich dual-arm peg-in-hole with model-free reinforcement learning or centralized dual-arm control for non-single object manipulation tasks in general.

3 LEARNING TO CENTRALIZE

In this section, we introduce our framework for dual-arm manipulation. We intend to reduce the required modeling effort to a minimum, which is why our approach is based on model-free reinforcement learning with sparse rewards. **Figure 2** illustrates our overall approach. In the first phase, we perform training in simulation. Our architecture includes a high-level policy outputting control targets (e.g., changes in end-effector position or joint angles, or additional changes in controller parameters). These targets are then followed by a lower-level policy represented by two hand-designed single-arm controllers. These controllers would then transform these control targets into joint torques, based on the robots' states and in certain cases using their dynamics models. In the second phase after successfully training the policy in simulation, we deploy it on the real-world setup without any further training in the real world (zero-shot transfer). This transfer requires a minor adaptation of the low-level controllers' parameters to ensure safety and smooth operation. The approach does not require a specific dual-arm controller since control is only coupled at a policy level.

3.1 Controller

The classical approach in centralized dual-arm control strategies is to model the manipulator's and the object's dynamics explicitly

and achieve coupling of both robotic arms by using multiple control loops. The system dynamics of the *i*th manipulator can be described by joint space equations of motion:

$$M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + g_i(q_i) = \tau_i - \tau_{ext_i} \quad (1)$$

$$\tau_{ext_i} = J_i^T(q_i)h_i \quad (2)$$

where q_i is the vector of joint positions, $M_i(q_i)$ is the inertia matrix, $C_i(q_i, \dot{q}_i)$ is the Coriolis/centrifugal matrix, $g_i(q_i)$ is gravity vector, τ_i is the vector of joint torques, and τ_{ext_i} represents external torques, which can be further decomposed into the external wrench h_i as in **(2)**. As both robots interact with each other directly or through manipulation of objects, the respective external wrenches have to include all forces and moments, which are applied to the robot. Hence, to create a valid dynamic chain, the rigid-body dynamics of the *j*th object are described by the following equation:

$$M_j\ddot{x}_j + C_j(\dot{x}_j)\dot{x}_j + g_j = h_j - h_{ext_j} \quad (3)$$

where M_j is the inertia matrix, C_j is the Coriolis/centrifugal matrix, g_j is the gravity vector, h_j the wrench exerted on the environment, and h_{ext_j} is the external wrench exerted by a, e.g., manipulator. In case that manipulators and objects form a closed dynamical chain, the system dynamics can be described by concatenating the respective external wrenches. Thereby, the dynamic equations need to be adapted to the specific manipulation task, but are not universally valid. For example, single object manipulation with two manipulators can be described as follows:

$$h_{object} \approx G_1 h_{robot1} + G_2 h_{robot2} \quad (4)$$

Whereas, dual-arm peg-in-hole could be defined by

$$\begin{gathered} h_{peg} \approx G_1 h_{robot1} \\ h_{hole} \approx G_2 h_{robot2} \\ ContactDynamicsModel\left(h_{peg}, h_{hole}\right) \end{gathered} \quad (5)$$

with G_i as respective grasp matrices, and *ContactDynamicsModel* as placeholder for contact modeling and of course with strongly simplified assumptions (e.g., neglecting mechanical stresses, no environment interactions, assuming force closure, neglecting object geometries, etc.).

Under considerations of simplified assumptions and various constraints, centralized single object manipulation can be tackled by classical control: Based on the dynamics model, commonly a hierarchical control strategy through multiple loops is applied, where the outer loops realize the main objective such as desired object movements and the inner loop accounts to generate a firm grasp and bounded internal forces (Caccavale and Villani, 2001; Caccavale et al., 2008; Erhart et al., 2013; Heck et al., 2013; Ren et al., 2015; Bjerkeng et al., 2014). The particular control loops can utilize any control strategy. Nevertheless, impedance control is a common choice to achieve compliant behavior (Caccavale and Villani, 2001; Caccavale et al., 2008; Heck et al., 2013; Ren et al., 2015), as contact forces are limited by coupling the control torque with position p and velocity v (**Eq. 6**). The control torque is calculated by multiplying the gains K_p and K_v with the difference of desired and actual position and velocity, respectively:

$$\tau = f_0\left(K_p(p_{des} - p) + K_v(v_{des} - v)\right) + f_1(p, v) \quad (6)$$

The principle can be applied in joint space or task space. f_0 and f_1 are generic functions to account for the variety of control laws and their additions (e.g., **Eq. 8**). Besides the success in dual-arm cooperative single object manipulation tasks, to our knowledge, no method exists for solving general assembly tasks with centralized control. Methods to explicitly model contacts and complex geometries are so far not powerful enough to formulate closed-loop dynamics and use a centralized controller. Even if control by classical methods would be feasible, an explicit set of dynamic equations, constraints, and adapted control loops for each task would be required. Hence, a general framework for dual-arm robotic manipulation needs to be based on a different approach.

An alternative way to bridge the gap is to use learning-based methods. Especially learning through interactions with the environment as in deep RL provides a promising way to facilitate complex manipulation tasks without the need for labeled data or human demonstrations. A learning signal is solely provided by rewarding the intended behavior. Deep RL then tries to maximize the accumulated reward, leading to policies/controllers that are compliant with the incentivized behavior.

Our approach is based on the idea of combining classical control with deep RL: We merge a policy network as high-level control and two independent low-level controllers. We thereby dispose of the need of designing a coupled control method in the classical sense. The controllers can be designed in a straightforward way without the need for purpose-built dual-arm control algorithms, allowing the use of any single-arm action space. The policy learns to inherently compensate for the constraints resulting from dual-arm interactions and provides individual action inputs for each controller. Furthermore, coupling at policy level is convenient to implement as the policies action space can simply be extended to include a second controller. The overall system is illustrated in **Figure 2**.

Besides the so far mentioned advantages of our framework, the method enables improved sim-to-real transfer: Both low-level controllers can be adjusted to the specifics of the real-world, whereas the high-level policy can be zero-shot transferred from simulation. Thereby, the need for expensive and difficult real-world experiments can be reduced to a minimum and the benefits from classical methods and learning-based approaches are combined.

3.2 Action Space

Classical control approaches for manipulation are often based on impedance control, as it comes with the previously mentioned advantages. However, since our method tries to compensate for interaction forces at a trajectory level, we

explore different control laws as action spaces and study their effect on the task success.

Joint Position Control

First of all, we use joint position control (**Eq. 7**) to compute a torque command: Both gains, k_p and k_v, are set to a constant value, q_{actual} and $\dot{q}_{actual}$ evaluated at run-time and $q_{desired}$ inferred by the policy.

$$\tau = k_p \cdot \left(q_{desired} - q_{actual}\right) + k_v \cdot \dot{q}_{actual} \tag{7}$$

Cartesian Impedance Control

Second, we implement Cartesian impedance control (Ott, 2008): The action space allows to move control from joint space to Cartesian space and includes model information such as the Cartesian inertia matrix $\Lambda(x)$ and the Jacobian matrix $J(q)$ as well as the gravity compensation term τ_{gc}. As the degrees of freedom exceed the number of joints, nullspace compensation τ_{ns} is added. Instead of $x_{desired}$, $\Delta x = x_{desired} - x_{actual}$ is directly passed in as action input.

$$\tau = J(q)^T \Lambda(x) \cdot \left(k_p \cdot \Delta x + k_v \cdot \dot{x}_{actual}\right) + \tau_{gc} + \tau_{ns} \tag{8}$$

Variable Cartesian Impedance Control

Variable Cartesian impedance control (Martín-Martín et al., 2019) is based on classical Cartesian impedance control, although adds k_p to the action space making control more variable to react with higher or lower stiffness if needed. We use anisotropic gains and couple velocity gains via $k_v = 2\sqrt{k_p}$ to achieve critical damping.

3.3 Reinforcement Learning

In our method, the policy is responsible for generating high-level trajectories, which are later on tracked by the chosen controller. We train the policy network using model-free RL. The policy receives the robot states to infer a control signal (action) for the aforementioned control laws (action spaces). We combine joint positions q_i, joint velocities $\dot{q}_i$, and joint torques τ_i of both robotic arms, respectively, as well as Cartesian positions and orientations of the end-effectors as state input $s = [q_0, \dot{q}_0, \tau_0, ee_{pos_0}, ee_{ori_0}, q_1, \dot{q}_1, \tau_1, ee_{pos_1}, ee_{ori_1}]$. Nevertheless, the state might need to be adjusted if the framework is applied for a different task, which for instance includes additional objects.

The proposed method is not restricted to a specific model-free RL algorithm, although an off-policy algorithm is desirable to facilitate high sample efficiency and allows the use of experience replay. Thus, we use Soft Actor-Critic (SAC) (Haarnoja et al., 2018) as the algorithm presents state-of-the-art performance and sample efficiency, but could potentially be replaced by others such as Deep Deterministic Policy Gradients (DDPG) (Lillicrap et al., 2016) or Twin Delayed Deep Deterministic Policy Gradients (TD3) (Fujimoto et al., 2018).

To enhance sample efficiency, the environment is implemented in a goal-based way. Thereby, the achieved goal $goal^{achieved}$ is returned alongside the environment state and can easily be compared with the desired goal $goal^{desired}$ to compute a reward r. Reward engineering is not necessary as we use a sparse reward (9).

$$r = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} \left| goal_i^{achieved} - goal_i^{desired} \right| < \delta \\ 0, & \text{else} \end{cases} \tag{9}$$

In Zuo et al. (2020), a similar setup is combined with the concept of Hindsight Experience Replay (HER) (Andrychowicz et al., 2017) for the single-arm robotic manipulation tasks push as well as pick-and-place. Their results point out that HER is sufficient to enhance the performance if only sparse rewards are available. Hence, to counteract the more challenging training task when using sparse compared with dense rewards, we use HER to augment past experiences. By replaying experiences with goals that have been or will be achieved along a trajectory, the agent shall generalize a goal-reaching behavior. Hence, unsuccessful experiences still help to guide an agent, as a sparse reward otherwise does not provide feedback on the closeness to the desired goal.

3.4 Implementation

We implement the general training and evaluation procedure in the following way: During each epoch, one full episode is gathered by interacting with the environment followed by 1,000 optimization steps. Moreover, we calculate the success rate every fifth epoch by averaging over 10 test cycles. We use ADAM (Kingma and Ba, 2017) as optimizer with a learning rate of 1e − 5 and a batch size of 256. The experience replay memory size is set to $800k$ and training starts after storing 10,000 samples. The q-network and policy-network consist of 4 and 3 linear layers, respectively, with a hidden dimension of 256 and ReLU (Agarap, 2018) activation functions. A visualization of both networks can be found in **Supplementary Figures S2, S3**. To update the target networks, we set an updating factor of 0.005. HER is set to use the strategy "future" with sampling of 6 additional experiences (Andrychowicz et al., 2017). All hyper-parameters are tuned manually and kept fixed for all experiments.

4 EXPERIMENTAL SETUP

We design experiments to answer the following questions:

- Can a central policy successfully learn dual-arm manipulations skills based on a decentralized control architecture?
- What action space leads to the highest success rate and the most robust policies?
- Is our method robust against disturbances and position uncertainty?
- Is a modular design enough to zero-shot transfer policies from simulation to the real world?

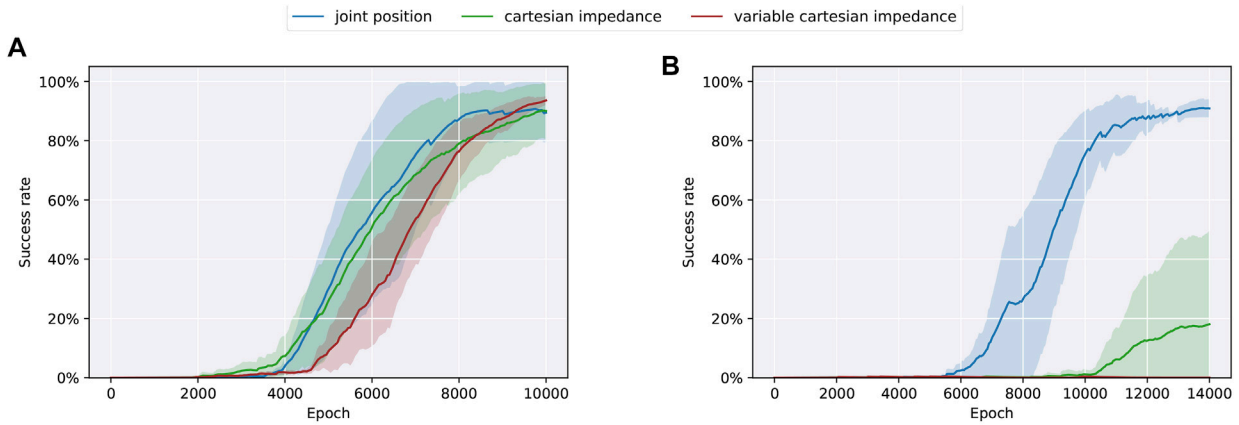

FIGURE 3 | Training results in simulation using different action spaces (joint position control, Cartesian impedance control, and variable Cartesian impedance control) on the task of dual-arm peg-in-hole assembly separated for different clearances [2 mm **(A)** and 0.5 mm **(B)**].

To answer these questions, we evaluate the proposed method on the task of peg-in-hole assembly with two Franka Emika panda manipulators both in simulation (**Figure 1A**) and in the real world (**Figure 1B**). The simulation environment is created using PyBullet (Coumans and Bai, 2016). We design it to match the real-world setup as closely as possible. Both setups are similar except for the environment frequency, which is 240 Hz in simulation and 1 kHz in the real world. The policy is operating at 60 Hz. Nevertheless, the robots only obtain the last state for control. To exclude the process of gripping and restrict movements of peg and hole, both are fixed to the end-effector of the respective robot arm. In the real-world experiments, this corresponds to the peg and hole being attached to the gripper of each manipulator. Furthermore, to enable an evaluation with increasing difficulty, pairs of pegs and holes have been created with a clearance of 2 and 0.5 mm. Moreover, we define the goal state as the relative distance between the positions of peg and hole. Both robots start with a significant distance and varying orientation with an offset around the initial joint position of q_{init} = [0.0, −0.54, 0.0, −2, −0.3, 3.14, 1.0]. We restrict robot movements by their joint limits, whereas the workspace is not bounded. Furthermore, the robot bases are positioned on a table with a distance of 1.3 m and both oriented to the same side. Respective stiffness values for each action space and applied joint limits can be found in the **Supplementary Tables S1, S2**. We use PyTorch (Paszke et al., 2019) to implement the models and train them on a single workstation equipped with an NVIDIA GeForce RTX 2080 GPU.

5 SIMULATION RESULTS

We use the simulation to train the policy as well as to perform ablation studies and robustness tests. As can be seen in the **Supplementary Video S1**,[1] the policy can be trained in simulation to learn a successful peg-in-hole insertion strategy. In addition to the video, **Supplementary Figure S1** shows that both manipulators move toward each other without any bigger diversion. The upper row in **Supplementary Figure S1** displays the achieved goals for all control variants. Two phases are clearly distinguishable: In the first phase, the end-effector's move quickly toward each other and in the second phase exact alignment and insertion takes place. All graphs in the second and third row display the sum of changes of end-effector positions and orientations and sum of changes of end-effector wrenches, respectively. It can be seen that both manipulators are involved in the process of aligning the end-effectors and pushing the peg inside. Hence, the approach does not lead to any leader/follower behavior where one end-effector just keeps its position similar to a single-arm solution. Although that is not the focus of this work, it is important to note that, when desired, our approach could theoretically lead to a leader/follower behavior by introducing an additional reward function incentivizing and punishing large actions taken by the leader and follower robots, respectively.

We design the experiments to start with an offset of ±10% from the initial joint positions and average all results over 4 seeds. All successful runs up to a clearance of 0.5 mm converge to a success rate above 90% in-between 10,000 and 14,000 epochs (**Figure 3**). As we use sparse rewards, no intermediate information about the task success is available. Hence, the success rates of a single run tend to converge either to 100% or stay at 0%, which can be easily seen for Cartesian impedance control and a clearance of 0.5 mm, where only 1 out of 4 runs converges in the given time frame. Overall, variance and training duration is increasing with smaller clearances, which confirms that the task becomes more challenging as clearances decrease.

Since our approach can be seen as an alternative to classical control methods and vanilla RL solutions, comparisons with both methods would be conceivable: Starting with vanilla RL solutions, we experimented with end-to-end policies that infer joint torques from states. As the policies did not converge to any successful

[1]https://sites.google.com/view/dual-arm-assembly/home

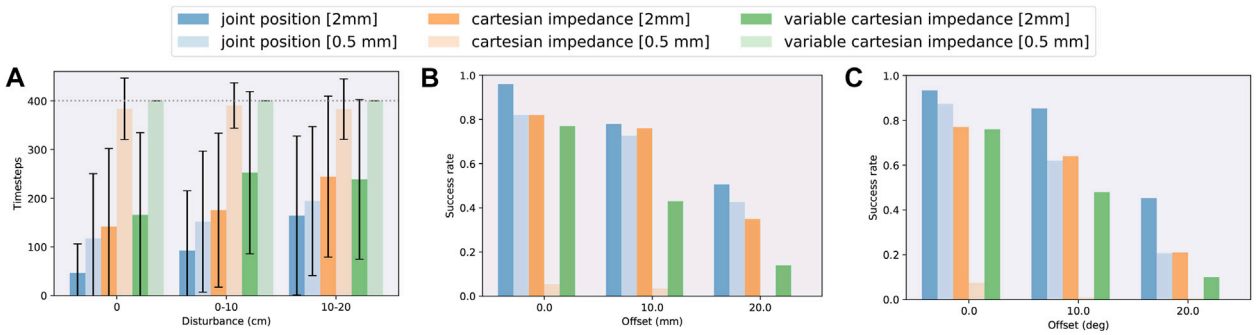

FIGURE 4 | Robustness when using different action spaces (joint position control, Cartesian impedance control, and variable Cartesian impedance control) on the task of dual-arm peg-in-hole assembly in simulation separated for different clearances. **(A)** Average episode durations when applying a random disturbance to the peg during a fixed time frame in the beginning of each episode. **(B)** Average success rates when applying a random position offset to the peg. **(C)** Average success rates when applying a random orientation offset to the peg.

behavior, leading to a success rate of 0% in all experiments, we refrain from comparing our approach with this class of solutions. As for classical control methods, we mention in the introduction and related work sections how these approaches are not task agnostic, limited to leader/follower behaviors, or require extensive modeling of interaction dynamics. Our method leads to a policy where both robots are involved and does not require an interaction dynamics model. Hence, because of different assumptions and solutions, a comparison with classical control methods is not feasible.

5.1 Ablation Studies

We compare the following control variants to study the effect of different action spaces: joint position control, Cartesian impedance control, and variable Cartesian impedance control as introduced in **Section 3.2**. **Figure 3A** shows the results when using a clearance of 2 mm, where policies trained in all three action spaces converge in a comparable manner. Moreover, to analyze the effect of smaller clearances, we conduct the same experiments using a 0.5-mm clearance between peg and hole (**Figure 3B**). However, only joint position control converges in the respective time frame. Overall, the results are different to Varin et al. (2019), where Cartesian impedance control converges faster than joint position control for single-arm peg-in-hole, and Martín-Martín et al. (2019), where variable Cartesian impedance control performs best in contact-rich manipulation tasks.

As peg-in-hole manipulation requires stiffness adaption, variable impedance control should theoretically perform best among the evaluated action spaces. In our experiments, this is only the case in the 2-mm environment, but does not seem to persist when the clearance is decreased. We suspect that this is due to the increased size of the action space, which makes learning the task more challenging, but could be partially decreased by introducing isotropic gains. In the case of Cartesian impedance control, we suppose that the under-performance could be attributed to the sub-optimal stiffness values. Utilizing more complex decentralized controllers comes with the downside that large effort is required for optimal parameter tuning. Hence, the results point out that even though our framework alleviates the need to model the coupling of both manipulators manually, both decentralized controllers still require specific system knowledge. Thus, future work should investigate ways to separate the stiffness adaptation from the learning process. That way, we could have sample efficient learning while keeping the stiffness values variable.

5.2 Robustness

First, we showcase the robustness by evaluating disturbance recovery, and second, we demonstrate the robustness against positioning errors.

5.2.1 Disturbance Recovery

To investigate the robustness to unforeseen events, such as collision or active manipulation by humans, we evaluate the success rate after being disturbed from time step 10 until 40, resulting in an end-effector offset. **Figure 4A** shows the results. Each episode duration, with a maximum of 400 steps, is averaged over 60 random disturbances, 3 test cycles, and all seeds. Afterwards, we compare their trajectories with a reference and calculate the disturbance as the difference between end-effector positions. Comparing all action spaces, it turns out that in our framework all variants can recover from external disturbances. Joint position control yields faster success, but episode durations increase proportionately more with higher external disturbances. Overall, the ability to recover depends mostly on the available time frame; hence, increasing the maximum time steps could allow handling larger disturbance offsets. **Figure 5** visualizes trajectories of desired joint positions given by the policy network when using joint position control. Two time steps after the external disturbance are applied; the policy starts to adapt to the external influence, varying on the disturbance magnitude. These results show that the policy network is actively reacting to the disturbance and not purely relying on the controller.

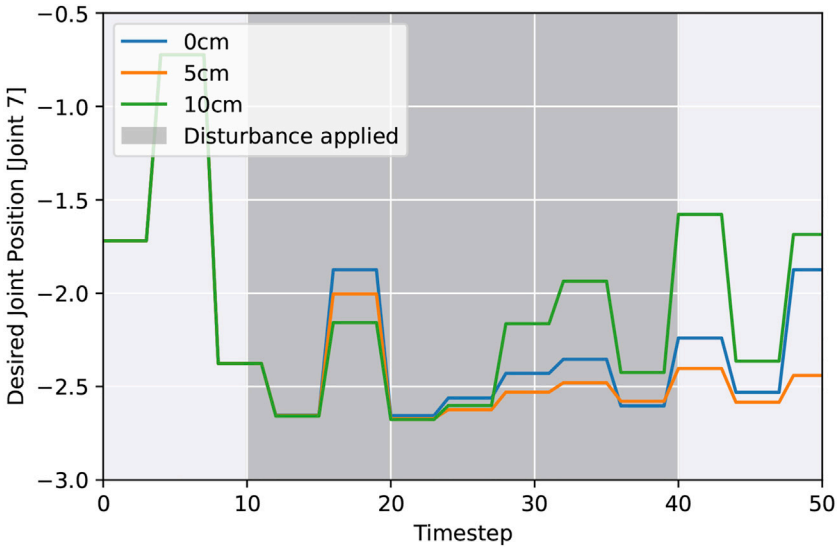

FIGURE 5 | Desired joint positions as inferred by the policy network (joint position control) after applying a random external disturbance.

5.2.2 Position and Orientation Uncertainties

Furthermore, to show that the agent does not simply learn the underlying kinematics, but a peg insertion behavior, we evaluate the performance under position and orientation uncertainties. As peg and hole positions are fixed during all training runs, the agent could learn to exploit the kinematics. Such a behavior is undesirable for sim-to-real transfer, as slight differences between simulation and real world would make a successful deployment impossible. Hence, by introducing offsets during testing, we can show the opposite. Moreover, position and orientation uncertainties will naturally occur in the real world, for instance, caused by grasping the peg in an earlier phase or inaccuracies during calibration. The given offset is the resulting value, which is randomly distributed in all directions, and is applied as the relative location of the peg to the end-effector. **Figures 4B,C** show the success rates for three position and orientation offsets, respectively. We average each result over 50 different offsets and all seeds. In general, the success rates decrease with higher offsets and smaller clearances. Cartesian impedance control turns out to be less robust compared with joint position control and variable Cartesian impedance control ends up last, which is comparable with the previous robustness evaluation of disturbance recovery. Nevertheless, joint position control and Cartesian impedance control are capable to handle large position offsets up to 10 mm with high success rates, which should already be sufficient for most applications and is significant considering that no randomization has been applied during training. Orientation offsets are even more challenging as they also introduce a position offset at the top of the peg. Still, joint position control has high success rates up to an offset of 10°. The evaluation under position and orientation uncertainties shows that the policy does not simply learn the underlying kinematics since peg and hole positions and orientations are fixed during training, but infers a peg insertion behavior based on the state information.

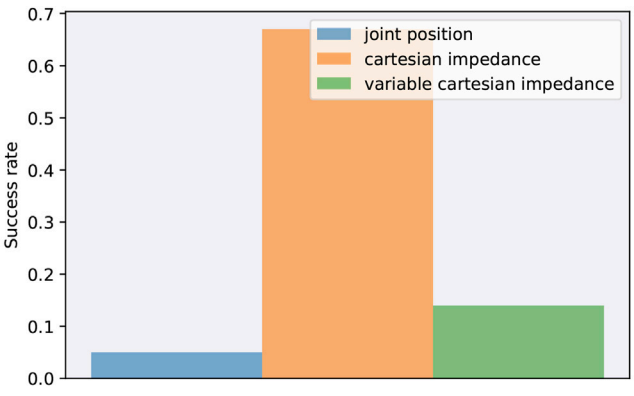

FIGURE 6 | Success rates when transferring a policy from simulation to reality with a random deviation of 5% from the initial joint positions q_{init}.

5.3 Real-World Results

To evaluate the approach in the real world, we transfer the policies trained in simulation without taking further measures to improve transferability such as domain randomization or domain adaption. We explicitly evaluate the performance without any additional steps targeting sim-to-real transfer, to precisely investigate if the policy is robust enough to be applied in reality and both decentralized controllers can compensate to bridge the gap between simulation and reality.

To enable zero-shot transfer of the approach, the simulation environment has been adapted to match the real world as close as possible. However, to ensure safe deployment in real-world

experiments, additional torque, velocity, and stiffness limitations need to be applied to establish joint limits and avoid large forces or high velocities. The taken precautions to guarantee observability and non-critical behavior come with the downside of further increasing the sim-to-reality gap and thereby affecting the final performance. All stiffness values and joint limits can be found in **Supplementary Tables S1, S2**. For all real-world experiments, we use a 3D printed peg and hole (material: polylactic acid), with a size of 26 mm × 26 mm × 30 mm and 30 mm × 30 mm × 32 mm, respectively. We choose a clearance of 2 mm since all action spaces have been successfully trained in simulation when using that clearance.

Figure 6 shows the success rates for each action space when starting with a random deviation of 5% for each joint from the initial joint positions q_{init}. Using Cartesian impedance control leads by far to the highest success rate as the action space helps to compensate for the sim-to-reality gap and is robust to the applied parameter changes for safe operation. Variable impedance control confirms the results of previous robustness evaluations as the variant cannot reach a high success rate in real-world evaluations. One reason for the performance decrease might be that the variable stiffness led to overfitting to the system dynamics in simulation instead of learning a generalized stiffness adaption. Joint position control, which performed best in simulation, is not able to keep up in reality at all. The action space is not robust to joint torque and joint velocity limitations, thus would require additional retraining using the applied limitations. Overall, the results show that a well-chosen action space can help to enhance robustness and transfer the approach from simulation to reality without applying further methods to target sim-to-real transfer. Moreover, the modular design helped to incorporate adaptions after training the policy, which would not have been possible in an end-to-end approach. Nevertheless, the proposed method leaves room for improvements: Among them, the impact of domain randomization and domain adaption should be explored in the future as well as fine-tuning in the real world to adapt the policy to the additionally applied safety constraints.

6 CONCLUSION

We introduce a framework for dual-arm assembly with the goal to compensate for constraint and interaction modeling of traditional centralized control. The approach explores the applicability of reinforcement learning by utilizing a policy network to couple decentralized control of both robotic arms without any explicit modeling of their interaction. The policy is trained through model-free reinforcement learning and can be combined with various well-established single-arm controllers. As we aim to explore a framework with a minimal set of task-specific assumptions, we only use sparse rewards. We evaluate the approach in simulation on the task of dual-arm peg-in-hole and show that joint position control provides good results up to an investigated clearance of 0.5 mm. Furthermore, we point out that in simulation the approach can recover from external disturbances and prove that the method learns a general peg insertion behavior by evaluating position uncertainties. Lastly, we zero-shot transfer the policy trained in simulation to the real world and show that a well-chosen action space can help to overcome the sim-to-reality gap. The framework can be seen as a first step in the direction of reducing modeling efforts for dual-arm manipulation and leaves lots of room for further research including the investigation of methods to improve sim-to-reality transferability and the evaluation of further dual-arm manipulation tasks. Moreover, sample efficiency needs to be enhanced for higher precision tasks such as peg-in-hole with smaller clearances; therefore, we plan to further optimize exploration, and experience replay and action spaces in the future.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**; further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

All authors contributed to the literature review, planning, idea creation, and writing tasks related to the paper. MA was the main person involved in the implementation and execution of experiments under the supervision and support of EA.

FUNDING

We greatly acknowledge the funding of this work by Microsoft Germany, the Alfried Krupp von Bohlen und Halbach Foundation, and project KoBo34 (project number V5ARA202) by the BMBF (grant no. 16SV7985).

ACKNOWLEDGMENTS

We would like to thank Carlos Magno C. O. Valle, Luis F. C. Figueredo, Karen Errou, Konstantin Ritt, Maximilian Ulmer, and Sami Haddadin for their general support and comments on this work.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2022.830007/full#supplementary-material

REFERENCES

Agarap, A. F. (2018). Deep Learning Using Rectified Linear Units (Relu). *ArXiv abs/1803.08375*.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., et al. (2017). Hindsight Experience Replay. *CoRR abs/1707.01495*.

Beltran-Hernandez, C. C., Petit, D., Ramirez-Alpizar, I. G., and Harada, K. (2020). Variable Compliance Control for Robotic Peg-In-Hole Assembly: A Deep Reinforcement Learning Approach. *CoRR abs/2008.10224*. doi:10.3390/app10196923

Beyret, B., Shafti, A., and Faisal, A. A. (2019). "Dot-to-dot: Explainable Hierarchical Reinforcement Learning for Robotic Manipulation," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE), 5014–5019. doi:10.1109/iros40897.2019.8968488

Bischoff, B., Nguyen-Tuong, D., Lee, I., Streichert, F., and Knoll, A. (2013). "Hierarchical Reinforcement Learning for Robot Navigation," in Proceedings of The European Symposium on Artificial Neural Networks, Computational Intelligence And Machine Learning (ESANN 2013).

Bjerkeng, M., Schrimpf, J., Myhre, T., and Pettersen, K. Y. (2014). "Fast Dual-Arm Manipulation Using Variable Admittance Control: Implementationand Experimental Results," in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014 (IEEE), 4728–4734. doi:10.1109/IROS.2014.6943235

Caccavale, F., Chiacchio, P., Marino, A., and Villani, L. (2008). Six-DOF Impedance Control of Dual-Arm Cooperative Manipulators. *Ieee/asme Trans. Mechatron.* 13, 576–586. doi:10.1109/TMECH.2008.2002816

Caccavale, F., and Villani, L. (2001). "An Impedance Control Strategy for Cooperative Manipulation," in IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 1, 343–348. doi:10.1109/aim.2001.936478

Clavera, I., Held, D., and Abbeel, P. (2017). "Policy Transfer via Modularity and Reward Guiding," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE), 1537–1544. doi:10.1109/iros.2017.8205959

Coumans, E., and Bai, Y. (2016). *Pybullet, a python Module for Physics Simulation for Games, Robotics and Machine Learning*.

Dietterich, T. G. (2000). Hierarchical Reinforcement Learning with the Maxq Value Function Decomposition. *Jair* 13, 227–303. doi:10.1613/jair.639

Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S., et al. (2020). *An Empirical Investigation of the Challenges of Real-World Reinforcement Learning*.

End, F., Akrour, R., Peters, J., and Neumann, G. (2017). "Layered Direct Policy Search for Learning Hierarchical Skills," in 2017 IEEE International Conference on Robotics and Automation (ICRA) (IEEE), 6442–6448. doi:10.1109/icra.2017.7989761

Erhart, S., Sieber, D., and Hirche, S. (2013). "An Impedance-Based Control Architecture for Multi-Robot Cooperative Dual-Arm mobile Manipulation," in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3-7, 2013 (IEEE), 315–322. doi:10.1109/IROS.2013.6696370

Fujimoto, S., van Hoof, H., and Meger, D. (2018). Addressing Function Approximation Error in Actor-Critic Methods. *CoRR abs/1802.09477*.

Ghavamzadeh, M., and Mahadevan, S. (2003). "Hierarchical Policy Gradient Algorithms," in *Computer Science Department Faculty Publication Series*, 173.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., et al. (2018). Soft Actor-Critic Algorithms and Applications. *CoRR abs/1812.05905*.

Heck, D. J. F., Kostic, D., Denasi, A., and Nijmeijer, H. (2013). "Internal and External Force-Based Impedance Control for Cooperative Manipulation," in European Control Conference, ECC 2013, Zurich, Switzerland, July 17-19, 2013 (IEEE), 2299–2304. doi:10.23919/ecc.2013.6669163

Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., and Levine, S. (2021). How to Train Your Robot with Deep Reinforcement Learning: Lessons We Have Learned. *Int. J. Robotics Res.* 40, 698–721. doi:10.1177/0278364920987859

Jain, D., Iscen, A., and Caluwaerts, K. (2019). "Hierarchical Reinforcement Learning for Quadruped Locomotion," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE), 7551–7557. doi:10.1109/iros40897.2019.8967913

Kingma, D. P., and Ba, J. (2017). *Adam: A Method for Stochastic Optimization*.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end Training of Deep Visuomotor Policies. *J. Machine Learn. Res.* 17, 1334–1373.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2016). "Continuous Control with Deep Reinforcement Learning," in 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016. Editors Y. Bengio and Y. LeCun.

Martín-Martín, R., Lee, M. A., Gardner, R., Savarese, S., Bohg, J., and Garg, A. (2019). Variable Impedance Control in End-Effector Space: An Action Space for Reinforcement Learning in Contact-Rich Tasks. *CoRR abs/1906.08880*.

Mehta, B., Diaz, M., Golemo, F., Pal, C. J., and Paull, L. (2020). "Active Domain Randomization," in Conference on Robot Learning (PMLR), 1162–1176.

Ott, C. (2008). *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*. 1 edn.. New York, NY, USA: Springer Publishing Company. Incorporated.

Pairet, È., Ardón, P., Broz, F., Mistry, M., and Petillot, Y. (2019). Learning and Generalisation of Primitives Skills towards Robust Dual-Arm Manipulation. *arXiv preprint arXiv:1904.01568*.

Park, H., Kim, P. K., Bae, J.-H., Park, J.-H., Baeg, M.-H., and Park, J. (2014). "Dual Arm Peg-In-Hole Assembly with a Programmed Compliant System," in 2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 431–433. doi:10.1109/URAI.2014.7057477

Parr, R., and Russell, S. (1998). Reinforcement Learning with Hierarchies of Machines. *Adv. Neural Inf. Process. Syst.* 1998, 1043–1049.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). "Pytorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32* (Red Hook, NY, USA: Curran Associates, Inc.).

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. (2018). "Sim-to-real Transfer of Robotic Control with Dynamics Randomization," in 2018 IEEE international conference on robotics and automation (ICRA) (IEEE), 3803–3810. doi:10.1109/icra.2018.8460528

Petitti, A., Franchi, A., Di Paola, D., and Rizzo, A. (2016). "Decentralized Motion Control for Cooperative Manipulation with a Team of Networked mobile Manipulators," in 2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016. Editors D. Kragic, A. Bicchi, and A. D. Luca (IEEE), 441–446. doi:10.1109/ICRA.2016.7487164

Ren, Y., Zhou, Y., Liu, Y., Jin, M., and Liu, H. (2014). "Adaptive Object Impedance Control of Dual-Arm Cooperative Humanoid Manipulators," in Proceedings of the World Congress on Intelligent Control and Automation (WCICA), 2015-March, 3333–3339. doi:10.1109/WCICA.2014.7053267

Schoettler, G., Nair, A., Luo, J., Bahl, S., Ojea, J. A., Solowjow, E., et al. (2019). Deep Reinforcement Learning for Industrial Insertion Tasks with Visual Inputs and Natural Rewards. *CoRR abs/1906.05841*.

Suomalainen, M., Calinon, S., Pignat, E., and Kyrki, V. (2019). Improving Dual-Arm Assembly by Master-Slave Compliance. *CoRR abs/1902.07007*. doi:10.1109/icra.2019.8793977

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 23–30. doi:10.1109/IROS.2017.8202133

Tsiamis, A., Verginis, C. K., Bechlioulis, C. P., and Kyriakopoulos, K. J. (2015). "Cooperative Manipulation Exploiting Only Implicit Communication," in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015 (IEEE), 864–869. doi:10.1109/IROS.2015.7353473

Varin, P., Grossman, L., and Kuindersma, S. (2019). "A Comparison of Action Spaces for Learning Manipulation Tasks," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, SAR, China, November 3-8, 2019 (IEEE), 6015–6021. doi:10.1109/IROS40897.2019.8967946

Vecerík, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., et al. (2017). Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards. *CoRR abs/1707.08817*.

Wang, Z., and Schwager, M. (2014). "Multi-robot Manipulation without Communication," in Distributed Autonomous Robotic Systems - The 12th International Symposium, DARS 2014, Daejeon, Korea, November 2-5, 2014. Editors N. Y. Chong and Y. Cho (Springer), 135–149. vol. 112 of Springer Tracts in Advanced Robotics. doi:10.1007/978-4-431-55879-8_10

Zhang, X., Zheng, Y., Ota, J., and Huang, Y. (2017). Peg-in-hole Assembly Based on Two-phase Scheme and F/T Sensor for Dual-Arm Robot. *Sensors* 17, 2004. doi:10.3390/s17092004

Zhu, H., Yu, J., Gupta, A., Shah, D., Hartikainen, K., Singh, A., et al. (2020). "The Ingredients of Real-World Robotic Reinforcement Learning," in ICLR 2020 Conference Blind Submission, 1–20.

Zuo, G., Zhao, Q., Lu, J., and Li, J. (2020). Efficient Hindsight Reinforcement Learning Using Demonstrations for Robotic Tasks with Sparse Rewards. *Int. J. Adv. Robotic Syst.* 17, 172988141989834–13. doi:10.1177/1729881419898342

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Alles and Aljalbout. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.