Reinforcement Learning of Impedance Policies for Peg-in-Hole Tasks: Role of Asymmetric Matrices

Shir Kozlovsky, Elad Newman, and Miriam Zacksenhouse

Abstract— **Robotic manipulators are playing an increasing role in a wide range of industries. However, their application to assembly tasks is hampered by the need for precise control over the environment and for task-specific coding. Cartesian impedance control is a well-established method for interacting with the environment and handling uncertainties. With the advance of Reinforcement Learning (RL) it has been suggested to learn the impedance matrices. However, most of the current work is limited to learning diagonal impedance matrices in addition to the trajectory itself. We argue that asymmetric impedance matrices enhance the ability to properly correct reference trajectories generated by a baseline planner, alleviating the need for learning the trajectory. Moreover, a task-specific set of asymmetric impedance matrices can be sufficient for simple tasks, alleviating the need for learning variable impedance control. We learn impedance policies for small (few mm) peg-in-hole using model-free RL, and investigate the advantage of using asymmetric impedance matrices and their space-invariance. Finally, we demonstrate zero-shot policy transfer from the simulation to a real robot, and generalization to new real-world environments, with larger parts and semi-flexible pegs.**

Index Terms—**Compliance and impedance control, machine learning for robot control, force and tactile sensing, reinforcement learning.**

I. Introduction

INDUSTRIAL robotic manipulators are becoming increasingly vital in modern manufacturing businesses. However, their application to contact-rich assembly tasks is hampered by the need for precise control over the location of the assembled items, substantially increasing the cost of the overall system [1]. Impedance and admittance control, have been demonstrated to enhance the ability to interact with the environment and handle uncertainties in location [2]–[6], but have to be properly tuned for each task.

Reinforcement learning (RL) provides a powerful tool for learning control polices. Given the importance of impedance control for assembly tasks, it has been suggested to learn the impedance matrices implicitly or explicitly, thus alleviating the need for tuning the impedance parameters for each task [7]–[9]. This is especially important for small and medium size industries where a variety of tasks are performed in small batches.

Focusing on manipulation tasks, it has been shown that learning the impedance in the Cartesian end-effector (EEF) space outperforms learning the impedance in the joint space and facilitates policy transfer [7]. However, learning impedance in the Cartesian space is usually restricted to diagonal matrices. We argue that asymmetric impedance matrices in the Cartesian space enhance the ability to perform assembly tasks, and demonstrate, for the first time, the advantage of learning asymmetric rather than symmetric impedance matrices.

RL includes model-based and model-free algorithms. Model-based RL algorithms are less suitable for contact-rich assembly tasks since it is difficult to model the interaction with the environment accurately. Model-free RL includes off-policy and on-policy algorithms. Off-policy algorithms seek to reuse previous experience, but are brittle and cannot be guaranteed to converge in continuous state and action spaces [10]. On-policy algorithms are guaranteed to converge and can be less brittle, but suffer from sample inefficiency since each gradient step requires new samples.

To overcome sample inefficiency, and to facilitate the transition from simulation to the physical world, our approach: (1) reduces the action-space and simplifies the policy by learning impedance matrices that depend only on task specifications (e.g., hole location), (2) specifies the impedance in the Cartesian EEF space, and (3) performs residual learning, i.e., the learned impedance policy is used to modify the reference trajectory generated by a baseline planner after initial contact, while a standard PD controller is used to follow the reference trajectory in free space. Thus, there is no need to learn the trajectory itself. While specifying the impedance in the Cartesian space is common, item (3) and especially item (1) are novel features of our approach. Learning only the impedance matrices has also the promise of facilitating the transfer from simulations to real robots.

Focusing on our approach, we address the following research questions, either in simulation ((1)–(4)) or on a real robot ((5)-(6)): (1) How well can RL learn impedance policies for small (few mm) peg insertion despite location uncertainties, even-thought the forces and especially the torques are small?

Manuscript received 24 February 2022; accepted 29 June 2022. Date of publication 15 July 2022; date of current version 23 August 2022. This letter was recommended for publication by Associate Editor J. Malzahn and Editor C. Gosselin upon evaluation of the reviewers' comments. The work was supported by the Israel Innovation Authority as part of the ART (Assembly by Robotic Technology) consortium under Grant 74885. *(Corresponding author: Shir Kozlovsky.)*

Shir Kozlovsky and Elad Newman are with the Autonomous Systems Program, Technion, Haifa 7403757, Israel (e-mail: shirkozlovsky@gmail.com; eladnwm@gmail.com).

Miriam Zacksenhouse is with the Faculty of Mechanical Engineering, Technion, Haifa 32000, Israel, and also with the Technion Autonomous Systems Program, Technion, Haifa, Israel (e-mail: mermz@technion.ac.il).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3191070, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3191070

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

(2) How robust is the learned impedance policy to new types of uncertainties (e.g., in orientation) that were not included in training? (3) What is the contribution of asymmetric impedance matrices to performance? (4) Are the parameters of the learned impedance matrices invariant to space? (5) How well does the learned policy transfer to real robots without retraining? (6) How well does the learned policy generalize to other real world environmental conditions, e.g., different sizes and semi-flexible pegs like electric wires?

II. Related Work

In a series of seminal papers, Hogan explained the importance of Cartesian impedance control for successful interaction with the environment [2], [11], [12]. Impedance control endows the EEF with the desired impedance, e.g., stiffness, damping and inertia, which determines the desired trade-off between position and force control. Interestingly, humans can modify the stiffness of their hands by co-contracting their muscles or by changing the posture of the arm [2], [13].

In robotic applications impedance control is usually implemented in software rather than in hardware. The standard dynamic based impedance control introduced by Hogan [2] relies on an accurate dynamic model of the robot [3], [14] and thus may hamper sim2real. Instead we implemented position based impedance control, also known as admittance control, which modifies the reference trajectory in response to force and torque (F/T) measurements [3], [6].

Given the importance of impedance control in robotic manipulations, it has been suggested to learn the proper impedance implicitly or explicitly [7], [9], [15]. Luo et al.learned the impedance control implicitly, via a neural network that determines the trajectory given the state and F/T measurements [15]. A number of researchers learned variable impedance explicitly, in addition to the trajectory itself, either in the Cartesian EEF space [7], [8] or in the joint space [16]. Focusing on manipulation tasks, Martin-Martin et al.compared between learning impedance in the joint space versus EEF space, and concluded that the latter is superior [7]. In either case, in all these papers explicit learning of the impedance matrices was restricted to diagonal matrices.

The potential role of non-diagonal or asymmetric impedance matrices was considered only lately. Impedance matrices in the joint space, developed for legged locomotion tasks using risk sensitive optimal control with measurement uncertainties, included asymmetric elements [17]. Focusing on the Cartersian EEF space, Oikawa et al. designed pre-determined non-diagonal stiffness matrices and learned which one to apply during specific insertion tasks [18], [19]. In a previous paper from our group [9], full asymmetric stiffness matrices were learned explicitly for a residual admittance policy. The learned policy successfully inserted pegs of different shapes and sizes (in the range of 25–60 mm), handled uncertainties in hole location and peg orientation, generalized well to new shapes and transferred well from simulations to the real world. Thus, our previous work suggests that: (1) learning asymmetric impedance matrices can be sufficient for properly modifying the trajectories generated by a baseline planner, thus alleviating the need to learn the trajectory explicitly, and (2) learning impedance matrices that depend only on task requirements (e.g., size, location), independent of state, can be sufficient, thus alleviating the need to learn variable impedance parameters.

III. Reinforcement Learning of Impedance Policies

A. Impedance and Admittance Control

We designed and implemented admittance control in the six-degrees of freedom (6-DOF) Cartesian space of the EEF. Given F/T measurements, $Q \in R^6$, admittance control modifies the reference trajectory to satisfy the desired spring-mass-damper behavior at the EEF [3]–[5]:

$$Q = M(\ddot{x}_m) + C(\dot{x}_m - \dot{x}_{ref}) + K(x_m - x_{ref}) \qquad (1)$$

where $x, \dot{x}, \ddot{x} \in R^6$ denote the 6-dimensional position, velocity, and acceleration vectors of the EEF in Cartesian space, the indices $*_{ref}$ and $*_m$ refer to the reference trajectory and the modified trajectory, respectively, and $M, C, K \in R^{6\times6}$ are the desired inertia, damping, and stiffness, respectively. Eq. (1) was implemented in the 12-dimensional state space defined by the position and velocity $X = [x, \dot{x}]$ and converted to a discrete dynamical system. At each time step, the discrete dynamical system was evolved to compute the modified trajectory X_m given the reference trajectory X_{ref} and F/T measurements, Q. Finally, the modified trajectory X_m was followed by a standard PD controller with diagonal gain-matrices.

Thus, the admittance controller was implemented by two control loops, as depicted in Fig. 1: (1) an external loop that modified the reference trajectory X_{ref} given the F/T measurements Q and computed X_m according to the desired dynamic behavior in (1), and (2) an inner loop that followed the modified trajectory X_m using standard PD control with diagonal gain-matrices. The reference trajectory X_{ref} was generated by a baseline planner as a minimum jerk trajectory [20] to the desired location X_{des}, via an intermediate point a few mm above X_{des}. The reference trajectory included a short pause at the intermediate point to prevent additional position errors. X_{des} was generated by the physical simulation or the experimental application by adding an error to the actual hole location as detailed in Sections IV-B and V-A, respectively.

B. Action Space and Impedance Matrices

The action learned in this research determines the Cartesian impedance matrices: the stiffness matrix K, damping matrix C and inertia matrix M. Proper selection of action space has a significant impact on robustness, task performance, learning efficiency, and exploration [7]. Learning the impedance matrices is motivated by the insight that humans control the stiffness of the hand to affect the interaction with the environment. Given the impedance matrices, the admittance controller modifies the reference trajectory in response to the F/T measurements, alleviating the need to learn the trajectory. Furthermore, learning proper Cartesian impedance policies that handle uncertainties in simulation facilitates the transfer to real robots. Finally a

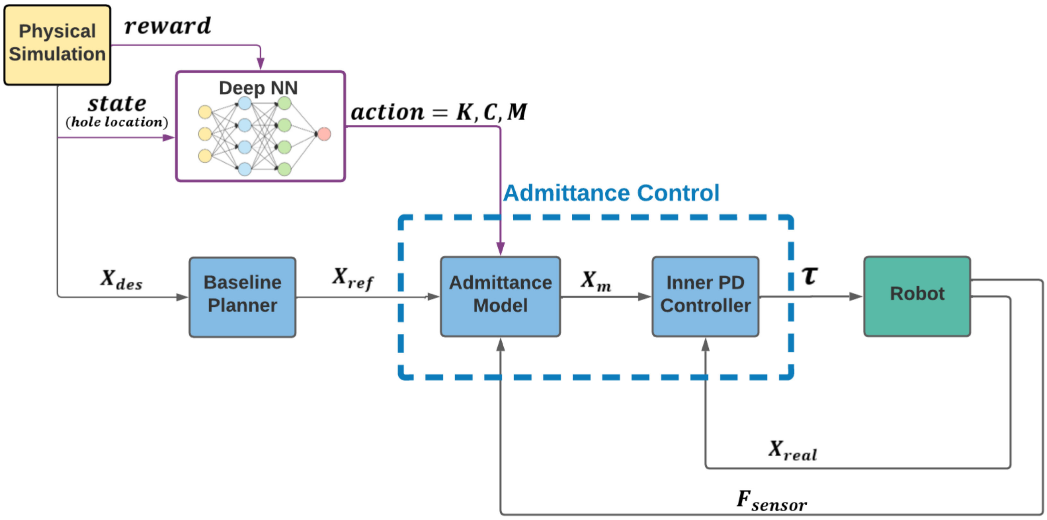

Fig. 1. Block Diagram of the overall system including the impedance policy learned using deep RL, and the admittance controller. X_{des} represents the erroneous hole location generated by the physical simulation. The baseline planner generates the reference trajectory X_{ref} as a minimum jerk trajectory to X_{des}. The impedance policy determines the desired impedance matrices (K, C, M). The admittance model modifies the reference trajectory to impose the desired impedance relationship and calculates the modified trajectory X_m. The inner PD controller calculates the joint torques τ required to follow X_m.

standard PD controller is applied to determine the joint torques needed to follow the modified trajectory, alleviating the need to implicitly learn the non-linear Jacobian.

We compared three types of Cartesian impedance matrices. In all cases, the inertia matrix was diagonal positive definite, while the stiffness and damping matrices were either: (1) diagonal, (2) non-diagonal symmetric, or (3) asymmetric matrices. We checked that the learned stiffness and damping matrices are positive definite (asymmetric matrices are positive definite if the symmetric part is positive definite [21], [22]). The use of asymmetric impedance matrices is motivated by the insight that peg-in-hole insertion can be facilitated by modifying the trajectory in the plane perpendicular to the axis of the hole in response to torques. The contribution of asymmetric Cartesian matrices to those modifications is demonstrated in Fig. 2 as explained next.

Matrices can be represented as the sum of a symmetric and anti-symmetric matrices. The symmetric part of an impedance matrix can be related to a potential function that yields a conservative force field, while the anti-symmetric part results in a curl field [12], [23]. Fig. 2 demonstrates the effects of the symmetric (upper panels) and anti-symmetric (lower panel) parts of the stiffness, K, and compliance, K^{-1}, matrices for a two dimensional case. The choice of the stiffness matrix was motivated by the results presented in Section IV-E. The stiffness matrix relates deviations in the x-axis, $p_x = x_1$, and in the angle around the y-axis, $\theta_y = x_5$, to the vector field representing the force along the x-axis, $F_x = Q_1$, and torque around the y-axis, $T_y = Q_5$ (left panels). The compliance matrix, K^{-1}, relates the F/T measurements F_x and T_y to required trajectory modifications p_x and θ_y, represented by the vector-field in the right panels.

Focusing on the effects of the compliance matrix, which is most relevant for admittance control, it is evident that the curl field generated by the anti-symmetric part contributes significantly to the ability to modify the trajectory along one axis, e.g., p_x, in response to forces or torques in another axis, e.g., T_y. This coupling is also generated by the symmetric part, but in that case the coupling is affected by the diagonal terms, which determine the shape of the equi-potential ellipsoids, and depends on the vector of applied forces and torques. In particular, the coupling vanishes when the diagonal terms are equal, so the ellipsoids

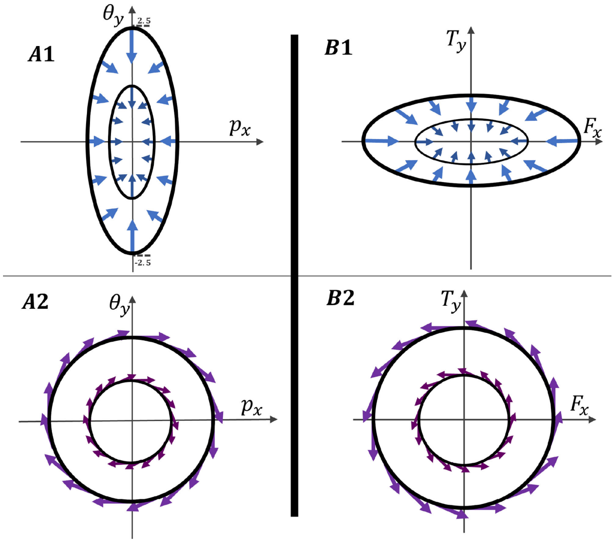

Fig. 2. The effect of symmetric (upper panels) and anti-symmetric (lower panels) of the stiffness matrix $K = [1\ 1; -1\ 2.5]$ (left panels) and the compliance matrix K^{-1} (right panels). The selected stiffness matrix was motivated by the mean stiffness matrix, μ_K, generated by one of the impedance policies studied in Section IV-E. In this example, the stiffness matrix relates deviations in the x-axis, $p_x = x_1$, and in the angle around the y-axis, $\theta_y = x_5$, to the vector field representing the force along the x-axis, $F_x = Q_1$, and torque around the y-axis, $T_y = Q_5$. The compliance matrix K^{-1} relates the F/T measurements F_x and T_y to the trajectory modifications p_x and θ_y represented by the vector-field in the right panels.

degenerate to circles, or when the vector of applied forces and torques is along one of the main axes of the matrix.

To limit the action-space, we restricted the stiffness and damping matrices to have only 4 off-diagonal parameters, which are expected to be most relevant for peg-in-hole tasks. Specifically, the stiffness and damping matrices have the form of the matrix A in (2), with the proper restrictions for the diagonal and symmetric cases:

$$A = \begin{bmatrix} A_{xx} & 0 & 0 & 0 & A_{xy_\theta} & 0 \\ 0 & A_{yy} & 0 & A_{yx_\theta} & 0 & 0 \\ 0 & 0 & A_{zz} & 0 & 0 & 0 \\ 0 & A_{y_\theta x} & 0 & A_{x_\theta x_\theta} & 0 & 0 \\ A_{x_r y} & 0 & 0 & 0 & A_{y_\theta y_\theta} & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{z_\theta z_\theta} \end{bmatrix} \quad (2)$$

C. Reinforcement Learning of Impedance Policies

As mentioned in Section I, we used model-free RL, since it is difficult to accurately model the interaction with environment. We implemented an on-policy algorithm, known as the proximal policy optimization (PPO, [24]), which facilitates convergence. The policy was implemented as a deep neural network (DNN) with two hidden layers of 32 neurons each and Leaky RelU activation functions using PyTorch [25]. The DNN received the 3-dimensional (erroneous) hole location as an input and determined the parameters of the impedance matrices. The number of parameters was 18, 22, and 26 for the diagonal, symmetric and asymmetric matrices, respectively. The action was determined once per episode, i.e., once per insertion trial, and the resulting impedance matrices were used throughout the trial.

The reward function included two parts: (1) Cumulative reward, and (2) Terminal reward. The cumulative reward was based on the relative state of the hole and peg at each step of the simulation. It included the cosine of the relative angle, the total and horizontal distances. The terminal reward was a constant of 40,000 points given only when the robot succeeded to insert the peg in the hole.

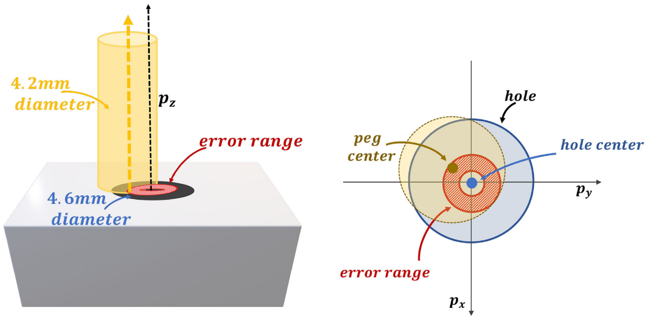

Fig. 3. Actual and erroneous hole locations. The center of the actual hole in the board is indicated by a blue dot. The baseline planner is given an erroneous estimate of the hole location X_{des}, uniformly distributed in a ring (red shadowed region) around the actual center of the hole with inner and outer radii of 0.21[mm] and 0.8[mm], respectively. The peg (yellow circle) is illustrated at a possible X_{des}. The diameters of the peg and hole are 4.2[mm] and 4.6[mm], respectively.

IV. Simulation

The simulation was designed to address the first four research questions raised in Section I. The basic research questions are about the ability to learn impedance policies for small peg-in-hole tasks and to handle uncertainties. The main research questions are about the contribution of the anti-symmetric part of the impedance matrices to performance and about space-invariance.

A. Simulation Architecture

The simulation was constructed based on Robosuite [26] using Mujoco physical engine [27], and includes four modules:

- RL module (detailed in Section III-C) includes the NN, which determines the action in the form of the impedance parameters (Eq.(2)). The action is determined only once every episode and sent to the Environment module.
- Environment module places the board in a random location. It adds a random error to the position of the hole, as detailed in Section IV-B, and sends the resulting desired location, X_{des}, to the robot module. The Environment module also computes the reward based on the state it receives from MuJoCo, as detailed in Section III-C, and sends the state and reward to the RL module.
- Robot module simulates a UR5e robot with F/T sensor. Initial tests were conducted to assure that similar forces are observed in both the simulation and the physical robot upon contact with the board under the same admittance controller. The robot module sends the action and X_{des} that it receives from the Environment to the Controller.
- Controller module implements both the baseline planner and the admittance controller depicted in Fig 1. The baseline planner generates the reference trajectory to X_{des} by computing a minimum jerk trajectory [20]. The admittance controller computes the vector of required joint torques, τ, as detailed in Section III-A and sends it to the Robot module.

B. Environmental Conditions

Simulations of peg insertion were conducted under the following conditions:

- Size: The diameters of the peg and hole were $D_p = 4.2$[mm] and $D_h = 4.6$[mm], respectively.
- Board location: The board was randomly located in the x, y plan, at the beginning of each episode, with uniform distribution in the working space.
- Location uncertainty: A random translation error was added to the actual center of the hole to generate the desired location X_{des} that was sent to the baseline planner. Errors were uniformly distributed in a ring with inner and outer radii of 0.21[mm] and 0.8[mm], respectively, as shown in Fig. 3, to assure overlap between the peg and the board.
- Orientation uncertainties: Orientation uncertainties were included only during testing. The orientation of the peg was uniformly distributed in a cone with apex angle of 24° around the z axis.

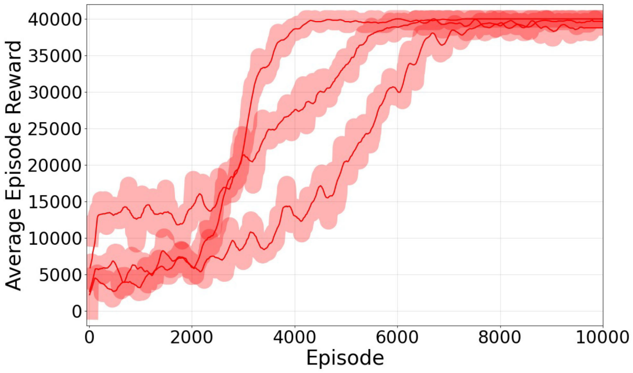

Fig. 4. Average episode reward during training of impedance policies with asymmetric matrices as a function of the number of episodes.

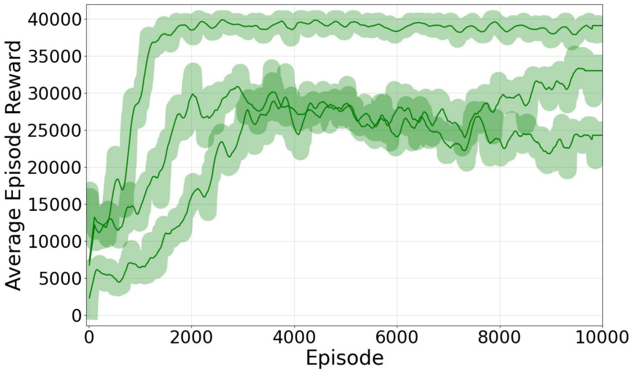

Fig. 5. Average episode reward during training of impedance policies with diagonal matrices as a function of the number of episodes.

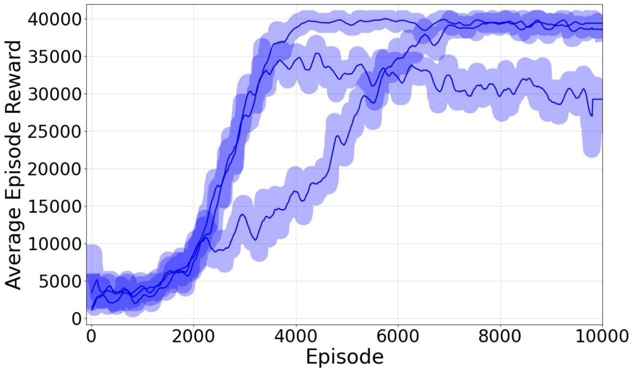

Fig. 6. Average episode reward during training of impedance policies with non-diagonal symmetric matrices as a function of the number of episodes.

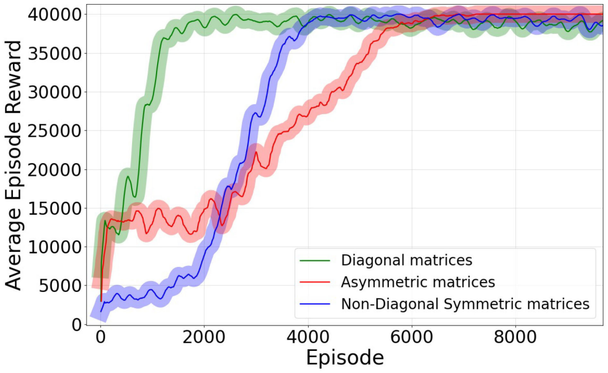

Fig. 7. Comparison of average episode reward during training sessions that resulted in best performance with diagonal, non-diagonal symmetric and asymmetric impedance matrices.

C. Performance With Asymmetric Impedance Matrices

Three types of impedance policies were compared, with three types of stiffness and damping matrices: (1) diagonal, (2) non-diagonal, and (3) asymmetric matrices, as detailed in Section III-B. Six training sessions were performed with each type of impedance policy to support statistical evaluation. Each training session was initialized with a randomly selected NN and seed. Training was conducted for 10000 trials and the best policy was saved for evaluation. This sub-section focuses on the performance of impedance policies with asymmetric matrices, while the next sub-section compares the performance of the different types of impedance policies.

Fig. 4 depicts average episode reward from 3 training sessions with asymmetric matrices. Reward grew rapidly after 2000–4000 episodes and converged after about 4000–6000 episodes reaching close to the maximum reward of 40,000. Performance of the best 6 policies, one from each of the 6 training sessions, were evaluated on 200 episodes with uncertainties in hole location. Success-rates ranged from 96–99.5%, with a mean of 98%.

As detailed in section IV-B, training was performed with uncertainties only in the position of the hole. To assess robustness to a new type of uncertainties, the policy that obtained a success-rate of 98.5% was also tested with uncertainties in both hole location and peg orientation. The resulting success rate was 97%, indicating that the policy generalizes well to new types of uncertainties that were not included in training.

D. Comparison of Different Types of Impedance Policies

Fig. 5 and Fig. 6 depict average episode reward from 3 training sessions with diagonal and non-diagonal symmetric matrices, respectively. Rewards obtained with diagonal matrices (Fig. 5) increased at earlier stages compared to rewards obtained with non-diagonal matrices (Fig. 4 and Fig. 6) but reached lower maximum levels. Most importantly, the maximum rewards reached with asymmetric matrices (Fig. 4) were higher than the rewards reached with symmetric matrices (Fig. 5 and Fig. 6).

Those differences are also evident in Fig. 7, which compares three representative training curves, one from each of the three types of matrices. The representative training curves are those that achieved the highest reward with the given type of impedance matrices. The early increase in the reward obtained when training with diagonal matrices can be attributed to the smaller dimension of the action-space (18 parameters compared to 22 and 26 for symmetric and asymmetric matrices, respectively). However, the small dimension of the action-space restricts the maximum reward that can be obtained with diagonal, and even non-diagonal symmetric matrices.

The superior performance of the trained policies with asymmetric impedance matrices is also apparent when comparing success rates. Specifically, success-rates obtained by the best 6 policies with diagonal matrices (one from each training session)

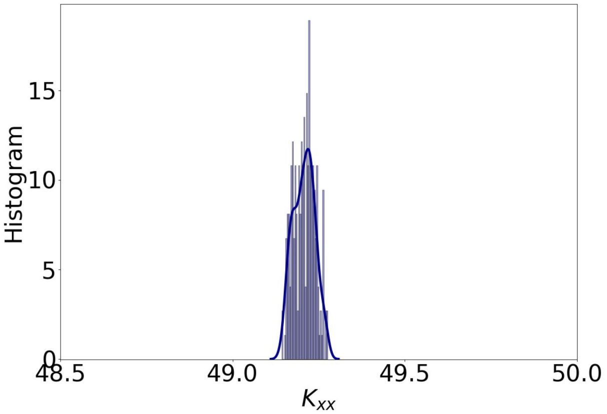

Fig. 8. Distribution of K_{xx}, the first element of the stiffness matrix, generated by one of the learned policies with asymmetric impedance matrices for 200 different hole locations.

ranged from 70–94.5% with a mean of 78%, while success-rates obtained by the best 6 policies with non-diagonal symmetric matrices ranged from 80–97% with a mean of 90%. Statistical analysis, conducted using Wilcoxon rank-sum test, indicates that the success-rates obtained by the best policies with asymmetric matrices (reported in Section IV-C) are significantly better than the success-rates obtained with diagonal matrices ($p = 0.004$) or with non-diagonal symmetric matrices ($p = 0.01$).

E. Space Invariance

The learned policy determines the set of impedance parameters as a function of the estimated location of the hole. However, space invariant policies can be advantageous, especially for real-world applications, reducing computation load and memory requirements. Since the admittance is defined in the Cartesian space of the EEF, it is hypothesized that space-invariant policies can provide similar performance to space-dependent policies. To verify this hypothesis, we estimated the distributions of the learned impedance parameters over space and evaluated the performance of a space-invariant policy.

Focusing on one of the learned policies with asymmetric impedance matrices, the distribution of each impedance parameter was computed over 200 different hole locations. The mean, μ, and standard deviation, σ, of the parameters of the stiffness K and damping C matrices are summarized in (3) and (4), respectively. The coefficient of variation ($C_v = \sigma/\mu$) of each parameter (including the parameters of the inertia matrix) is smaller than $1.2 \cdot 10^{-3}$, indicating that the distributions are narrow around the mean. This is also evident in Fig. 8, which depicts the distribution of the first element in the main diagonal of the stiffness matrix, K_{xx}.

$$\mu_K = \begin{bmatrix} 49.2 & 0 & 0 & 0 & 42.53 & 0 \\ 0 & 1.15 & 0 & 24.94 & 0 & 0 \\ 0 & 0 & 32.7 & 0 & 0 & 0 \\ 0 & -16.16 & 0 & 82.36 & 0 & 0 \\ -50.2 & 0 & 0 & 0 & 119.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 23.3 \end{bmatrix}$$

$$\sigma_K = \begin{bmatrix} 0.003 & 0 & 0 & 0 & 0.026 & 0 \\ 0 & 0.0007 & 0 & 0.016 & 0 & 0 \\ 0 & 0 & 0.02 & 0 & 0 & 0 \\ 0 & 0.009 & 0 & 0.025 & 0 & 0 \\ 0.031 & 0 & 0 & 0 & 0.073 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.014 \end{bmatrix} \quad (3)$$

$$\mu_C = \begin{bmatrix} 118 & 0 & 0 & 0 & -86.32 & 0 \\ 0 & 17.7 & 0 & -42.9 & 0 & 0 \\ 0 & 0 & 47.8 & 0 & 0 & 0 \\ 0 & 25 & 0 & 53.05 & 0 & 0 \\ -14.4 & 0 & 0 & 0 & 28.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 71.2 \end{bmatrix}$$

$$\sigma_C = \begin{bmatrix} 0.007 & 0 & 0 & 0 & 0.05 & 0 \\ 0 & 0.00011 & 0 & 0.02 & 0 & 0 \\ 0 & 0 & 0.02 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0.03 & 0 & 0 \\ 0.008 & 0 & 0 & 0 & 0.011 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.04 \end{bmatrix} \quad (4)$$

Note: the 2-dimensional stiffness matrix considered in Fig. 2 is approximately proportional to the 2-dimensional sub-matrix of μ_K: $[\mu_K(1,1)\ \mu_K(1,5); \mu_K(5,1)\ \mu_K(5,5)]$.

The narrow distributions of the impedance parameters generated by the learned policy over different hole locations suggest that a space invariant policy can be sufficient. This was investigated by evaluating the performance of a space-invariant policy, which generates the mean impedance matrices independent of hole location. The success rate was 98% - a minor degradation from the success rate of 98.5% obtained with space-dependent parameters over the same number of trials. Thus, space invariant policies can be sufficient.

V. Simulation to Real Transfer (Sim2Real)

Transferring policies that were learned in simulation to real robots, a step known as sim2real, usually requires re-training on the real robot. As mentioned above, we expect our method, which learns Cartesian impedance matrices that depend only on task specifications, to facilitate sim2real. Thus, we evaluated the performance of a policy that was learned in simulation on a real robot, without retraining. Furthermore, we evaluated the ability of the policy to generalize to new environments.

A. Sim2Real - Experimental Conditions

The performance of one of the policies with asymmetric matrices was evaluated on a physical robot, without any retraining. Experiments were conducted using the industrial cobot, UR5e, with OnRobot HEX-E F/T sensor. For comparison we also evaluated the performance of a PD controller with parameters that provided the best performance during hand-tuning peg-insertion experiments on the physical robot. The following

TABLE I
EXPERIMENTAL RESULTS WITH A UR5E COBOT: SUCCESS RATES AND THEIR 95% CONFIDENCE INTERVALS ESTIMATED FROM 200 (FIRST 5 LINES) OR 100 (LAST 2 LINES) TRIALS, AS DETAILED IN THE TEXT

Experiment type	Peg diameter [mm]	Hole diameter [mm]	Error size (radius) [mm]	PD controller		learned impedance policy	
				Success rate	Confidence interval	Success rate	Confidence interval
Basic performance	4.2	4.6	±(0.21 – 0.8)	68%	(61 – 74)%	**84**%	(78 – 88)%
Generalization & Evaluation	4.5	6	±2.5	60%	(53 – 67)%	**79**%	(72 – 84)%
	8.5	10	±2.5	68%	(61 – 74)%	**94**%	(88 – 97)%
	8.5	10	±3.5	49%	(42 – 56)%	**72**%	(65 – 78)%
	9.5	10	±2.5	25%	(19 – 31)%	**84**%	(78 – 88)%
Generalization to semi-flexible peg	4	6	±2.5	54%	(44 – 64)%	**81**%	(72 – 88)%
	4	connector	±2.5	21%	(15 – 30)%	**83**%	(74 – 90)%

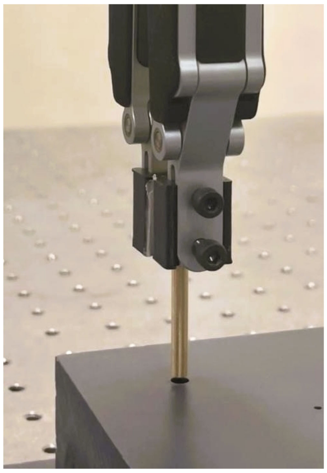

Fig. 9. Experimental set-up to evaluate basic performance: inserting a rigid 4.2[mm] diameter peg into a 4.6[mm] diameter hole with translation errors in a ring of ±(0.21–0.8)[mm].

7 experiments were performed with each controller, under the conditions specified in Table I:

- Experiment 1: Basic performance was evaluated under the exact same conditions as in simulation, inserting a rigid 4.2[mm] diameter peg into a 4.6[mm] hole with errors of ±(0.21–0.8)[mm], as shown in Fig. 9.
- Experiments 2-5: Generalization and evaluation of the effects of size, including larger pegs, errors and clearance, as detailed in Table I.
- Experiments 6-7: Generalization to semi-flexible pegs involving the insertion of a 4[mm] electrical wire with crimped terminal into either a 6[mm] diameter hole or a connector, as shown in Fig. 10. The cross-section of the hole in the connector was non-circular with 5–6[mm] diameter. Translation errors were $\pm(2.5)$[mm]. The gripper held the electrical wire just above the terminal.

Each of the first 5 experiments included 2 sets of 100 trials each, while the last 2 experiments included 1 set of 100 trials. Each set of 100 trials was conducted at a different location in space, and performed automatically. We note that multiple contacts with the surface during the automatic testing might have caused additional errors beyond the pre-set errors.

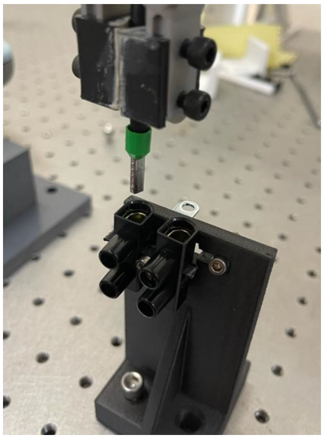

Fig. 10. Experimental system to evaluate generalization to semi-flexible pegs: inserting a 4[mm] electrical wire with crimped terminal into a connector with non-circular cross section 5–6[mm] diameter and ±2.5[mm] errors.

B. Sim2Real - Results

Success rates and their 95% confidence intervals are summarized in Table I for each of the experiments. Basic performance, reported in the first line, was evaluated under the same conditions as in simulation. The learned impedance policy achieved a success rate of 84%. While this is a good performance, it is below the success rate of 98.5% achieved by the selected policy in simulation. This discrepancy may be attributed to larger errors, beyond the planned ones, caused by multiple contacts with the surface during automatic testing, and to other differences between the simulation and physical world. Nevertheless, the relatively good performance indicates that despite those differences, sim2real transfer remains relatively effective.

In the interest of evaluating performance with larger errors, we next tested insertion of parts with a larger clearance. Inserting 4.5[mm] and 8.5[mm] pegs into holes with 1.5[mm] clearance (in diameter), the learned impedance policy achieved success rates of 79%, and 94%, respectively, despite translation errors of ±2.5%[mm] (2nd and 3rd lines). The better performance with the larger peg may be attributed to larger interaction forces and torques. As expected, success rate deteriorated with the size of the error as is evident by comparing the 3rd and 4th lines (with a 8.5[mm] peg).

Interestingly, performance with large parts remained high even with clearance of 0.5[mm]. Specifically, inserting a 9.5[mm] peg into a 10[mm] hole, the learned policy achieved a success rate of 84% (line 5). In this case the success rate of the PD controller was especially poor. This may be attributed to the difficulty of the task, which requires precise correction despite large area of contact, compared with other experiments.

Most importantly, the impedance policy generalized well to an industrial application involving the insertion of a 4[mm] diameter crimped terminal wire into either a 6[mm] diameter hole or a connector, as shown in Fig. 10. The learned impedance policy achieved over 80% success rate, much better than the below 55% success rate of the PD controller. Finally, we note that the impedance controller outperformed the PD controller in other experiments too, with non-overlapping 95% confidence intervals.

VI. Conclusion

This study proposes to learn Cartesian impedance policies for robotic control of assembly tasks, and demonstrates their success on small (few mm) peg-in-hole tasks. The policy determines the elements of asymmetric impedance matrices, as a function of hole location. A fixed set of impedance matrices is learned for each hole-location, and used to modify the reference trajectory after initial contact with the surface, thus facilitating peg insertion despite uncertainties in hole location. By simplifying the policy, reducing the action space and using residual learning, we overcome the problem of sample inefficiency of on-policy RL and facilitate sim2real.

We demonstrated the ability of the learned policies to handle uncertainties in hole location, which were included in training, and even uncertainties in peg orientation, which were not included in training. The learned policy was evaluated on a real robot, UR5e, without further re-training (single-shot policy transfer [28]), and successfully generalized to new environments, with either larger pegs and holes or crimped terminal wires and connectors.

The most significant contribution of our work is in demonstrating the advantage of learning impedance policies with asymmetric, rather than symmetric or diagonal matrices. Specifically, success rates obtained with asymmetric matrices were significantly better than those obtained with symmetric or diagonal matrices. Furthermore, the impedance parameters for different hole locations in the workplace were narrowly distributed, and a space-invariant policy achieved similar performance to space-dependent policy. Space-invariant policies can further simplify learning and industrial implementation.

References

[1] O. Kroemer et al., “A review of robot learning for manipulation: Challenges, representations, and algorithms,” *J. Mach. Learn. Res.*, vol. 22, pp. 1395–1476, 2021.

[2] N. Hogan, “Impedance control Part1-3,” *Trans. ASME, J. Dyn. Syst., Meas., Control*, vol. 107, no. Mar. 1985, pp. 1–24, 1985.

[3] T. Valency and M. Zacksenhouse, “Accuracy/robustness dilemma in impedance control,” *Trans. ASME, J. Dyn. Syst., Meas. Control*, vol. 125, no. 3, pp. 310–319, 2003.

[4] T. E. Milner, “Impedance control,” in *Encyclopedia Neuroscience.*, Berlin Heidelberg,Germany: Springer, 2009, ch. 6, pp. 1929–1934, 2008.

[5] R. Volpe and P. Khosla, “The equivalence of second-order impedance control and proportional gain explicit force control,” *Int. J. Robot. Res.*, vol. 14, pp. 574–589, 1995.

[6] M. Schumacher et al., “An introductory review of active compliant control,” *Robot. Auton. Syst.*, vol. 119, pp. 185–200, 2020, 2019.

[7] R. Martín-Martín, M. A. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg, “Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1010–1017.

[8] C. C. Beltran-Hernandez et al., “Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach,” *Appl. Sci. (Switzerland)*, vol. 10, no. 19, pp. 1–17, 2020.

[9] O. Spector and M. Zacksenhouse, “Learning contact-rich assembly skills using residual admittance policy,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 6023–6030.

[10] H. R. Maei et al., “Convergent temporal-difference learning with arbitrary smooth function approximation,” *Adv. Neural Inf. Process. Syst.* vol. 22, pp. 1204–1212, 2009.

[11] N. Hogan, “The mechanics of multi-joint posture and movement control,” *Biol. Cybern.*, vol. 52, no. 5, pp. 315–331, 1985.

[12] —, “Impedance control: An approach to manipulation: Part III—Applications,” *J. Dyn. Syst., Meas., Control*, vol. 107, pp. 121–128, 1985.

[13] H. Höppner et al., “Neural, mechanical, and geometric factors subserving arm posture in humans,” *Journ. Neurosci.*, vol. 5, pp. 2732–2743, 1985.

[14] A. Lu and Z.; Goldenberg, “Implementation of robust impedance and force control,” *J. Intell. Robot. Syst.*, vol. 6, no. 2, pp. 145–163, 1992.

[15] J. Luo et al., “Reinforcement learning on variable impedance controller for high-precision robotic assembly,” *IEEE Int. Conf. Robot. Automat.*, 2019, pp. 3080–3087.

[16] M. Bogdanovic, M. Khadiv, and L. Righetti, “Learning variable impedance control for contact sensitive tasks,” *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6129–6136, Oct. 2020.

[17] B. Hammoud, M. Khadiv, and L. Righetti, “Impedance optimization for uncertain contact interactions through risk sensitive optimal control,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4766–4773, Jul. 2021.

[18] M. Oikawa, K. Kutsuzawa, S. Sakaino, and T. Tsuji, “Admittance control based on a stiffness ellipse for rapid trajectory deformation,” in *Proc. IEEE 16th Int. Workshop Adv. Motion Control*, 2020, pp. 23–28.

[19] M. Oikawa, T. Kusakabe, K. Kutsuzawa, S. Sakaino, and T. Tsuji, “Reinforcement learning for robotic assembly using non-diagonal stiffness matrix,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2737–2744, Apr. 2021.

[20] T. Flash and N. Hogan, “The coordination of arm movements: An experimentally confirmed mathematical model,” *J. Neurosci.*, vol. 5, no. 7, pp. 1688–1703, 1985.

[21] M.C. Pease, *Methods of matrix algebra.* New York NY, USA: Academic, 1965.

[22] C. R. Johnson, “Positive definite matrices,” *Amer. Math. Monthly*, vol. 77, no. 3, pp. 259–264, 1970. [Online]. Available: https://www.jstor.org/stable/2317709

[23] E. Bizzi et al., “Regulation of multi-joint arm posture and movement,” *Prog. Brain Res.*, vol. 64, pp. 345–351, 1986.

[24] J. Schulman et al., “Proximal policy optimization algorithms,” *Comput. Res. Repository*, pp. 1–12, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[25] A. Raffin et al., “Stable-baselines3: Reliable reinforcement learning implementations,” *Journ. Mach. Learn. Res.*, vol. 22, pp. 1–8, 2021.

[26] L. Fan et al., “SURREAL: Open-source reinforcement learning framework and robot manipulation benchmark,” in *Proc. Conf. Robot Learn.*, 2018, pp. 767–782.

[27] E. Todorov, T. Erez, and Y. Tassa, “MuJoCo: A physics engine for model-based control,” in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.

[28] R. Kirk et al., “A survey of generalisation in deep reinforcement learning,” *Comput. Sci.*, pp. 1–43, 2022.