IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 July 2024, accepted 12 July 2024, date of publication 23 July 2024, date of current version 5 August 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3432644*

RESEARCH ARTICLE

# Robust Contact-Rich Task Learning With Reinforcement Learning and Curriculum-Based Domain Randomization

**ALI AFLAKIAN, JAMIE HATHAWAY, RUSTAM STOLKIN, (Member, IEEE), AND ALIREZA RASTEGARPANAH**
Extreme Robotics Laboratory, School of Metallurgy and Materials, University of Birmingham, B15 2TT Birmingham, U.K.

Corresponding author: Alireza Rastegarpanah (a.rastegarpanah@bham.ac.uk)

This work was supported in part by the "Research and Development of a Highly Automated and Safe Streamlined Process for Increase Lithium-Ion Battery Repurposing and Recycling" Project called (REBELION) under Grant 101104241, and in part by the U.K. Research and Innovation (UKRI) Project "Reuse and Recycling of Lithium-Ion Batteries" (RELIB) under RELiB2 Grant FIRG005 and RELIB3 Grant FIRG057.

**ABSTRACT** We propose a framework for contact-rich path following with reinforcement learning based on a mixture of visual and tactile feedback to achieve path following on unknown environments. We employ a curriculum-based domain randomisation approach with a time-varying sampling distribution, rendering our approach is robust to parametric uncertainties in the robot-environment system. Based on evaluation in simulation for compliant path-following case studies with a random uncertain environment, and comparison with LBMPC and FDM methods, the robustness of the obtained policy over a stiffness range $10^4$–$10^9$ N/m and friction range 0.1–1.2 is demonstrated. We extend this concept to unknown surfaces with various surface curvatures to enhance the robustness of the trained policy in terms of changes in surfaces. We demonstrate $\sim 15\times$ improvement in trajectory accuracy compared to the previous LBMPC method and $\sim 18\times$ improvement compared to using the FDM approach. We suggest the applications of the proposed method for learning more challenging tasks such as milling, which are difficult to model and dependent on a wide range of process variables.

**INDEX TERMS** Model predictive control, contact-rich task, reinforcement learning, domain randomization.

## NOMENCLATURE

| | |
|---|---|
| RL | Reinforcement learning. |
| MPC | Model predictive control. |
| LBMPC | Learning-based MPC. |
| FDM | Virtual forward dynamics model. |
| FDM-L | Low-gain FDM. |
| FDM-H | High-gain FDM. |
| DR | Domain randomization. |
| DDPG | Deep deterministic policy gradient. |
| TD3 | Twin delayed DDPG. |
| TCP | Tool center point. |
| RGBD | Red blue green and depth information. |
| LSTM | long short-term memory network. |
| RMSE | Root mean square error. |
| ReLU | Rectified linear unit. |

## I. INTRODUCTION

In recent years, modern robots equipped with advanced sensor arrays have emerged as versatile platforms for automating a wide range of manual and repetitive tasks through their ability to interact with their surroundings. These tasks often involve intricate interactions with physical objects and surfaces, collectively falling under the umbrella of "contact-rich" tasks. Contact-rich path following forms a subset of

The associate editor coordinating the review of this manuscript and approving it for publication was Ton Duc Do.

contact-rich tasks, where the aim is to follow a desired path in contact with a surface, typically while modulating contact forces. Examples of such tasks include robotic grinding, polishing, and cutting through various materials with precision, akin to skilled artisans using handheld tools to meticulously carve intricate designs into wood or stone. These tasks demand continuous contact and precise movement to achieve desired outcomes. Uncertainty, whether in the form of imprecise environmental models or variations in the properties of the objects being manipulated, presents a challenge in the realm of disassembly and decommissioning. For instance, in the disassembly of electric vehicle (EV) batteries, a robotic arm equipped with a cutting tool faces the challenge of accurately navigating deformable surfaces while modulating contact forces, amidst uncertainties such as variations in material properties and surface conditions. Such uncertainties can lead to deviations from planned trajectories due to imprecise environmental models or unexpected variations in object properties. Addressing these challenges requires to develop a robust control algorithm capable of modelling and adapting to unknown contact dynamics and environment changes.

As an overview, traditional control approaches such as hybrid position-force control encounter limitations due to the requirement for task specification, or decoupling the motion and force controlled directions into orthogonal and independently controlled subspaces. Similarly, approaches such as impedance control aim to decouple the problem of path following into individual problems of trajectory planning and imposing a desired closed-loop dynamic behaviour of the robot. It is necessary to specify or adapt the desired trajectory; that is, it is necessary not only to specify the desired path, but also velocities, for example, the speed of a polishing task, or the feed rate of a milling tool. To this end, many control issues may be formulated as optimal control problems with discrete-time dynamics and cumulative costs across time. One well-known technique to solve such optimal control problems is MPC which predicts the behaviour of a system based on its model. MPC employs an online optimization method to determine the best control action to converge the expected output to some desired reference trajectory [1]. However, MPC suffers from several well-known disadvantages, chiefly, high computational complexity, which impedes its real-time deployment [2]. Furthermore, MPC approaches demand substantial domain expertise or – in the case of LBMPC – extensive offline labeled training data, necessitating comprehensive coverage of the state and action space to ensure model accuracy. Moreover, the accuracy of the predictive model employed significantly impacts controller performance [3].

Reinforcement learning approaches show promise for addressing such issues since they allow agents to acquire behaviours through interaction with their surroundings and adapt to novel, previously unencountered scenarios [4]. In this paper, we focus on a specific contact-rich application – robotic cutting – as a representative example, in the context of disassembly of electric vehicle (EV) batteries. An illustrative example is the disassembly of a battery module, where the robot must precisely cut through a deformable surface while simultaneously maintaining contact and regulating contact forces, even when the properties of such environments are uncertain. However, this challenge extends beyond cutting and is a fundamental characteristic of various contact-rich tasks. We investigate the use of vision-guided RL for contact-rich path following with parametric uncertainties. In Figure 1, we provide a graphical overview of the training steps for the proposed MPC and RL methods.

TABLE 1. Key features comparison of selected methods in this study.

| Method | Requires tuning | Online adaptation | Processing complexity | Tracking accuracy |
|---|---|---|---|---|
| FDM | Gain(for each task) | no | Low | Medium |
| LBMPC | Cost (one-time) | yes | High | Medium |
| This work | Reward (one-time) | yes | Medium | High |

Our main contributions include a thorough comparison of vision-guided RL, MPC, high-gain FDM, and low-gain FDM controllers in terms of speed, and tracking accuracy. Additionally, we propose a new vision-guided RL algorithm that takes into account various surfaces and achieves satisfactory performance. Table 1 provides a comparison of the key features of each selected method in this study.

Overall, our study demonstrates the potential of vision-guided RL for addressing the challenges of contact-rich tasks with parametric uncertainties.

## II. RELATED WORKS

Several approaches exist for tackling the challenges presented by high sample complexity and the suitability of RL to real-world deployment, particularly concerning contact-rich tasks. One such approach is based on noting the complementary advantages and shortcomings of MPC and RL, using the former to act as an expert policy to assist the latter. This has been applied in [5] and [6]. In the latter, a combination of MPC with RL providing worst-case performance guarantees was proposed, enabling online deployment with improved learning stability. However, the baseline model used neglects the parametric uncertainty in the environment. This is typical of most explicit MPC approaches, which require considerable domain expertise and prior knowledge, necessitating approaches with higher computational complexity. LBMPC is one such approach that has been used to address the problem of uncertain environments in robotic manipulation tasks [7], [8]. In LBMPC, a model of the system is learned through interaction with the environment, and this model is then used to predict future system behaviour and generate control actions. For example, an LBMPC approach for contact-rich path following with reduced sample complexity was considered in [9], exploring the capability of memory-augmented neural networks as a system model in

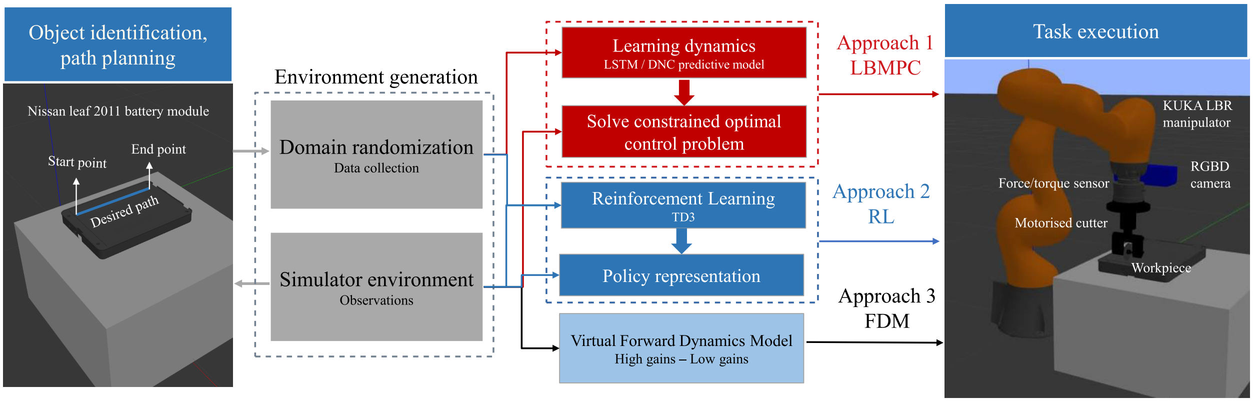


**FIGURE 1.** Graphical overview of the proposed MPC and RL methods for path following in contact-rich cutting task.

an MPC framework incorporating visual feedback. However, the high computational complexity of recurrent network architectures - particularly of memory-augmented neural networks such as the differentiable neural computer - limits the applicability for online deployment.

In a similar vein, recent works have emphasised the selection of suitable action spaces to facilitate learning of tasks in the real world. In [10], a task-frame formalism was employed to directly train a real-world policy for the task of vegetable cutting by exploiting the task-specific action-space constraints, however this approach is naturally dependent on accurate prior-knowledge of the task specification. In [11] and [12], the selection of suitable action spaces for contact-rich manipulation tasks is explored, with related approaches presented based on a variable impedance control system in which the controller gains constitute the policy action space. Methods based on this approach separate the task of compliant path following, which is essential for many interaction tasks, into the distinct problems of trajectory generation and trajectory tracking. Accurate task planning is not always possible to obtain and may not be desired in many applications as explored in [13]. Besides [13], task planning has been explored extensively in the context of MPC [14], [15], though demonstrably remains an issue for RL.

Many RL-based methods are developed in simulation, owing to the difficulty of transferring learned behaviours to the real world, particularly for destructive tasks. To address this, alternative approaches based on RL have been explored to narrow the sim-to-real transfer gap. The domain randomisation approach aims to close the reality gap by exposing the agent to a large number of scenarios that may not individually reflect the real-world system. However, it results in a policy representation that is robust to system uncertainties such that it may adapt to the real-world case. Examples of this approach include [16] for path following using an industrial robot. Curriculum-based, or automatic DR has been employed with success in dexterous manipulation tasks as demonstrated by OpenAI in [17] and [18], reducing the trial and error procedure of defining an environment with suitable variation to ensure a robust policy, while improving the capability to learn challenging tasks from scratch.

Our proposed approach builds upon some of these existing methods by combining TD3 with curriculum-based DR to learn contact-rich path following in a coupled robot environment system. We demonstrate the robustness and effectiveness of the learned policy representation to unknown environments, as validated through six random case studies in simulation, and comparing it with the virtual FDM controller and our earlier method in [9] that used LBMPC. We have also extended our work by randomizing height-field surfaces alongside randomizing stiffness and friction in our online training, which allowed us to generalize the trained policy for various types of surfaces beyond just planar surfaces.

Our work contributes to the growing body of research on using Deep RL for robotic manipulation tasks in uncertain environments and offers an approach for addressing the challenge of obtaining a robust policy representation with DR in complex tasks such as contact-rich manipulation tasks.

## III. MATERIALS AND METHODS

We consider a RL-based approach to the case of learning contact-rich path following for position-controlled manipulators in simulation based on the TD3 algorithm. For this work, we demonstrate this on the KUKA LBR iiwa R820 collaborative robot with an external force-torque sensor, without precluding the generality of the method to any position-controlled robot. The robot is equipped with an RGBD vision sensor mounted at the wrist and a contact cutting tool. For path following in contact with an environment, the desired behaviour is for the robot to track as closely as possible the desired path on the surface of an elastically compliant object with (potentially) unknown stiffness, while regulating or limiting the contact forces according to task requirements, and to avoid tool or workpiece damage.

Based on this specification, we consider the relevant raw measurements available as the TCP pose, $\boldsymbol{P}$ comprising position $\boldsymbol{p}$ and ZYX Euler angle orientation $\boldsymbol{R}$, and the external measured wrench $\boldsymbol{f_e}$. Let $\boldsymbol{r}(h)$ be the position vector of a path parameterized by arc length $h$. Then, the path direction at any point on the path can be calculated as:

$$\hat{\boldsymbol{c}}(h) = \frac{\boldsymbol{r}'(h)}{\|\boldsymbol{r}'(h)\|} \tag{1}$$

where $\boldsymbol{r}'(h)$ is the first derivative of the position vector with respect to arc length, and $\|\boldsymbol{r}'(h)\|$ is its magnitude. The path direction vector $\hat{\boldsymbol{c}}$ is a unit vector that points in the direction of the tangent to the path at the point $\boldsymbol{r}(h)$. For simplicity, and due to the natural constraints imposed by the geometry of the tool, we consider the case of a linear reference path that will be modified to match the geometry of the surface. The path is defined by a start- and end-point with position $\boldsymbol{p}_{\text{start}}, \boldsymbol{p}_{\text{end}}$ respectively. The path direction $\hat{\boldsymbol{c}}$ is then defined as

$$\hat{\boldsymbol{c}} = \frac{\boldsymbol{p}_{\text{start}} - \boldsymbol{p}_{\text{end}}}{\left|\left|\boldsymbol{p}_{\text{start}} - \boldsymbol{p}_{\text{end}}\right|\right|} \tag{2}$$

Although the TCP position is known directly, it is desirable to not expose its measurement directly to the agent to avoid over-fitting to specific tasks. Instead, the position is converted into a pair of task-specific features as the scalar distance ($s$) from the end point of the path:

$$s = \left(\boldsymbol{p} - \boldsymbol{p}_{\text{end}}\right) \cdot \hat{\boldsymbol{c}}, \tag{3}$$

and deviation from path $d$

$$d^2 = \left|\left|\boldsymbol{p} - \boldsymbol{p}_s\right|\right|^2 \tag{4}$$

The surface position estimate $p_s$ was computed as a Gaussian weighted average of the sampled points in the depth image about the closest point on the desired path to the current TCP position [9].

## IV. TASK 1: COMPARISON OF RL APPROACH WITH MPC AND FDM METHODS

The principle of the LBMPC approach [9] is to learn a model of the contact dynamics, given states, actions $\boldsymbol{x}$, $\boldsymbol{u}$ as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{5}$$

formulating the trajectory optimization as a constrained nonlinear optimization problem of some metric of cost, specified by $L$:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{J}(\boldsymbol{U}) = \sum_{i=0}^{N-1} L\left(\boldsymbol{x}_{k+i}, \boldsymbol{u}_{k+i}\right) \\ \text{s.t.} \quad & \boldsymbol{x}_{k+i+1} = \boldsymbol{f}(\boldsymbol{x}_{k+i}, \boldsymbol{u}_{k+i}) \\ & ||\boldsymbol{u}_{k+i}||_1 \leq u_{\max} \\ & i = 0, 1 \ldots N-1 \end{aligned} \tag{6}$$

where $||\cdot||_1$ denotes the $\ell^1$ norm. In the LBMPC approach, the function $\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ is represented as an LSTM neural network which is trained from trajectories collected offline.

To include an additional comparison method, we have adopted the use of an FDM for contact-rich Cartesian robot control, as detailed in the reference [19]:

$$\ddot{\boldsymbol{q}} = \mathbf{H}^{-1}\mathbf{J}^T\boldsymbol{f} \tag{7}$$

where $\mathbf{H}$ corresponds to the mass matrix of the robot, $\mathbf{J}$ is the Jacobian matrix, $\ddot{\boldsymbol{q}}$ represents the joint accelerations, and $\boldsymbol{f}$ is external force:

$$\boldsymbol{f} = \mathbf{K}_p\boldsymbol{e} + \mathbf{K}_d\boldsymbol{e}_d \tag{8}$$

while $\boldsymbol{e}$ is the distance error between the target and the current end-effector positions, $\boldsymbol{e}_d$ denotes the derivative of the distance error, $\mathbf{K}_p$ and $\mathbf{K}_d$ are positive definite diagonal stiffness and damping gains. The authors demonstrated that the FDM approach is not only free from delays and noise but also inherently more stable in contact-rich applications compared to traditional Admittance controllers [19].

We explored two distinct sets of gain values, high-gain (FDM-H) and low-gain (FDM-L). We defined $\mathbf{K_p}$ as follows:

$$\begin{aligned} \text{FDM-H:} \quad & \mathbf{K_p} = \text{diag}([100, 100, 1000, 10, 10, 10]) \\ \text{FDM-L:} \quad & \mathbf{K_p} = \text{diag}([10, 10, 200, 1, 1, 1]) \end{aligned}$$

In both methods, we used the following value for damping gains: $\mathbf{K_d} = \text{diag}([1, 1, 1, 0.1, 0.1, 0.1])$

TD3 is an off-policy actor-critic learning algorithm. Its principle of operation is related to the DDPG with key improvements in the introduction of twin critics, policy smoothing, Q-value clipping, and delayed actor updates [20]. It assumes the control problem can be modelled as a Markov decision process, in which the objective is to determine a policy that maximises an expected sum of rewards over time, weighted temporally by a discount factor. To ensure a fair comparison between the two methods, we design the reward function for the RL algorithm to be identical to the negated cost function used in the MPC approach. Hence:

$$L(\boldsymbol{x}, \boldsymbol{u}) = -r(\boldsymbol{x}, \boldsymbol{u}) \tag{9}$$

For path following, based on the objectives defined in Section III, we hence define the reward function $r$:

$$r\left(\boldsymbol{x}, \boldsymbol{u}\right) = -w_d d^2 - w_s \frac{|s|}{||\boldsymbol{c}||} - w_u \boldsymbol{u}^2 \tag{10}$$

where $w_d$, $w_s$, and $w_u$ are manually tuned weighting terms. The deviation term, represented by the expression $w_d d^2$, is a scaling penalty that discourages excessive deviations from the desired path. The $w_s \frac{|s|}{||\boldsymbol{c}||}$ term, referred to as the slicing term, encourages the agent to progress along the path. The normalisation by $||\boldsymbol{c}||$ ensures the reward for path progression is independent of the path length. This reward also encodes desirable traits like productivity; as the cumulative path progress penalty is minimised by agents that rapidly reach the path endpoint. $w_u \boldsymbol{u}^2$ is a small effort penalty to discourage extreme motions, labelled the effort term. For both approaches (MPC and RL), weighting contributions of $w_d = 1000$, $w_s = 10$, and $w_u = 0.000001$ were selected for each reward, which were selected to be equivalent to the setup in our previous work [9]. Through manual adjustment of these weights, we ensured they provide an optimal trade-off between the different objectives. As explained the deviation term penalizes deviations from the desired path. A higher

**TABLE 2.** The reinforcement learning hyperparameters and noise options used in training the TD3 policy.

| RL parameters | | Noise options | |
|---|---|---|---|
| Smooth factor | 0.001 | Mean | 0 |
| Mean attraction | 2.5 | Variance decay rate | 0.00001 |
| Sample time ($T_s$) | 0.02 | Variance | 0.5 |
| Discount factor | 0.99 | | |

weight on this term means the agent focuses more on minimizing path error, which can lead to the agent staying very close to the path but progressing slowly. On the other hand, if the slicing term has a lower weight, the agent may prioritize minimizing deviation over progressing along the path, resulting in inefficient movement along the path. Moreover, the effort term introduces a small penalty for extreme motions, promoting smoother and more controlled actions. This term has a very small weight to ensure it does not dominate the reward function but still discourages unnecessary movements.

### A. REINFORCEMENT LEARNING

Based on the available observations $\boldsymbol{x} = (d, s, \Delta\boldsymbol{p}, \boldsymbol{f}_e)$ we aim to learn a policy mapping $\boldsymbol{x}$ to actions in Cartesian velocity space $\boldsymbol{u}$. Each observation $\boldsymbol{x}$ was scaled to the approximate range 0–1. For TD3, we employ a set of deep feed-forward neural networks serving as the actor and dual critics respectively. The critic networks each comprise two input pathways: two hidden layers of 400 and 300 units for observations, and one of 300 units for actions, followed by a common output layer. For the critic network, a learning rate of $5 \times 10^{-4}$ was chosen to control how quickly the model updates its parameters based on the error at each step. Not to mention that a high learning rate would lead to instability, while a low rate would slow down learning. An L2 regularisation penalty of $2 \times 10^{-4}$ was selected to prevent overfitting, which helps the model work well on new data.

The actor network comprises 2 hidden layers of 400 and 300 units respectively. An initial learning rate of $5 \times 10^{-4}$, L2 regularisation penalty of $1 \times 10^{-5}$, ReLU hidden activation and tanh output activation were chosen as the network hyperparameters. The choice of ReLU helps the model to learn complex patterns (converting linear inputs into non-linear outputs) and tanh activation bounds the velocities by saturating the policy outputs. We used a batch size of 512, meaning the model updates its learning using 512 examples at a time. Larger batch sizes may stabilize learning but require more memory. We have also used a discount factor of 0.99 to value long-term rewards as much as immediate rewards. This encourages the model to make decisions that are beneficial in the long run. A sample time of 0.02 was selected to determine the frequency at which actions are sampled and executed, balancing the trade-off between computational load and control precision. The remaining hyperparameters for the learning algorithm were chosen according to Table 2. Most of these hyperparameters were initially chosen based on default and recommended values from the literature.

**TABLE 3.** Sample space used for domain randomisation of the simulated workpiece parameters.

| Property | Range | Distribution |
|---|---|---|
| Stiffness $k_p$ ($Nm^{-1}$) | $10^4$–$10^9$ | Log-uniform |
| Dyn. coeff. friction $\mu$ | 0.1–1.2 | Uniform |

From this baseline, we adjusted them through trial-and-error (manual search) to better fit our specific problem. Training was carried out up to a threshold of 3000 episodes. This threshold is established from initial experiments conducted in a simulation environment described in Section IV-B. Based on the actions $\boldsymbol{u}$ and sample time $T_s$, we convert the policy outputs into joint position commands $\boldsymbol{q}$ as:

$$\boldsymbol{q} = T_s \cdot \mathbf{J}^{+}\boldsymbol{u} \tag{11}$$

where $\mathbf{J}^{+}$ is the Moore-Penrose pseudo-inverse of the manipulator Jacobian.

### B. DOMAIN RANDOMISATION ENVIRONMENT

To learn a policy representation that is robust to an unknown environment, we establish the environment based on a curriculum-based domain randomisation method. During training, at the beginning of each episode, the properties of the object surface were sampled according to the distributions in Table 3. In the traditional DR, the distribution parameters are held constant from the first episode as the range specified in Table 3, denoted as $l_+$, $l_-$ for the maximum and minimum value of a variable $l$. However, the extreme and immediate variation in the environment can greatly increase the difficulty of learning the task, and in some cases reaching the optimal reward is not possible as the learning algorithm converges to a local minimum. To combat this, we introduce the concept of curriculum-based DR. Under this approach, the full random distribution range is not immediately introduced, but varied according to each episode as $F_N$:

$$F_N = F_0 + (1 - F_0)g(N) \tag{12}$$

where $F_0$ is the fraction of the limits at episode zero. The maximum and minimum limits for episode $N$, $l_{N+}$, $l_{N-}$, are computed as

$$l_{N\pm} = l_- + (1 \pm F_N)\,\frac{l_+ - l_-}{2} \tag{13}$$

$g(N)$ is an envelope function that specifies the evolution of the randomisation distribution over the training process. In this study we select $g(N)$ as a linear function of $N$ as:

$$g(N) = \frac{N}{N_{max}} \tag{14}$$

### C. EXPERIMENTS

We evaluate the trained agents in the simulation environment discussed in Section IV-B, for the task of compliant path following along the surface of a given workpiece. Training for the TD3 agent with the curriculum DR approach was

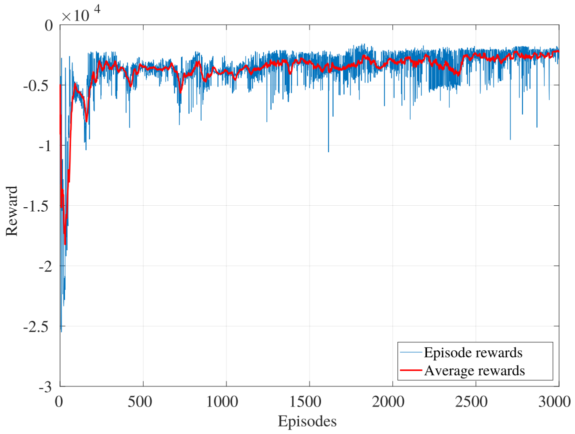


**FIGURE 2.** The training graph of the TD3 agent for the compliant path following task on a near-planar surface with unknown surface properties..

carried out over 4.05 × $10^5$ seconds for 3000 total episodes. The DR hyperparameter $F_0$ was chosen as 0.05 to allow some early variation in the material parameters to mitigate embedding of behaviours early in the training process that are overly dependent on a single material, while $N_{max}$ was chosen as 2500 based on the number of training episodes, allowing 500 episodes of training with the full randomisation range. The processor of the computer used for simulation and training was an Intel(R) Core(TM) i7-8086K 8-core processor with 4 GHz base clock and 32 GB RAM. The training graph is shown in Figure 2. The agent rapidly converges to an average reward of approximately −2500 and remains close to this value which illustrates that the desired task behaviour was successfully learned.

Based on the learned policy representation from the curriculum DR method, we evaluate the performance of the agent over six path following case studies with randomly chosen surface properties, shown in Table 4. For comparison, we employ a method based on LBMPC described in our previous work [9], with data collected using a series of manually designed admittance controllers to train a predictive model of the surface contact dynamics. Due to the difficulty of solving for the optimal trajectory $\boldsymbol{U}$ directly, we employ the forward shooting method using sample-based optimisation to approximate the solution of (6). For LBMPC a dataset of 101120 samples was collected, which is comparable to the number of observations exposed to the agent after ∼200 episodes of training. The average reward displayed in Figure 2 demonstrates that the agent experiences the majority of its performance improvement before completing 200 episodes of training. This is notable because the dataset size used for LBMPC consists of roughly the same number of observations as those encountered by the agent in this early phase of training, indicating that the dataset is a reasonable size for comparison of LBMPC and RL methods. This establishes a benchmark that is less sample intensive than the exploration required for RL but has greater computational overhead. The choice of case studies 3 to 6 is to study the behaviour of the RL and MPC when encountering stiffness and friction values outside of the defined ranges in the DR. The point is to evaluate how well RL and MPC generalize to situations they have not specifically been trained on.

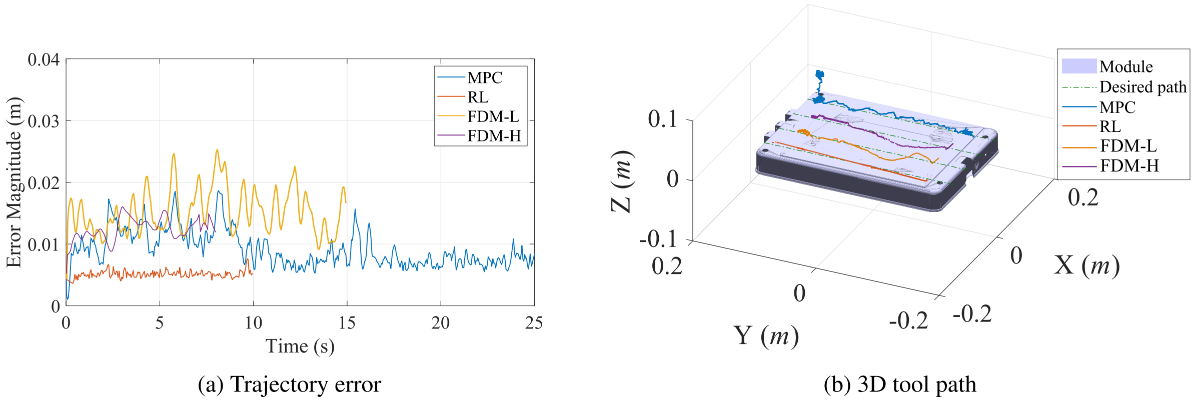
**TABLE 4.** The stiffness and friction coefficients used in different case studies. Bold data are defined outside the training range for domain randomization.

| **Experiments** | **Stiffness $k_p$ (Nm$^{-1}$)** | **Friction $\mu$** |
|---|---|---|
| Case study 1 | $9.22116 \times 10^7$ | 0.95413 |
| Case study 2 | $2.19674 \times 10^6$ | 0.797441 |
| Case study 3 | $5.06996 \times 10^5$ | **1.32379** |
| Case study 4 | $9.79874 \times 10^8$ | **1.40105** |
| Case study 5 | $\mathbf{5.00000 \times 10^3}$ | 0.66567 |
| Case study 6 | $\mathbf{2.00000 \times 10^{10}}$ | 0.41343 |

The magnitude of tracking error and cutting path for LBMPC, TD3 with curriculum DR, FDM-H, and FDM-L during each example case study are presented in Figure 3–8. Figures 3a–8a illustrate the magnitude of trajectory errors, represented as the norm of the error vector in the $x$, $y$, and $z$ coordinates, which captures the difference between the current and desired tool TCP positions.

In Figure 3a, for case study 1, the task was completed in approximately 25 seconds using MPC, 10 seconds with the trained RL agent, 15 seconds with the FDM-L method, and 6 seconds with the FDM-H method. This comparison highlights that RL achieves faster task execution compared to both MPC and the FDM-L approach. However, the trajectory error of the RL method is significantly lower than that of the other three methods. This suggests that the RL agent is more effective at completing the task compared to the MPC and FDM methods. Figure 3b displays the 3D path of the tool-tip for case study 1, comparing the performance of the RL method with others. Although all methods exhibit attempts to correct any deviation from the path, the deviation is less noticeable for the RL method. This observation is supported by the RMSE between the end-effector position and the desired path. The RMSE was 7.2mm for MPC, 9.6mm for FDM-L, and 9.0mm for FDM-H, greatly exceeding the corresponding RL value of 0.56mm.

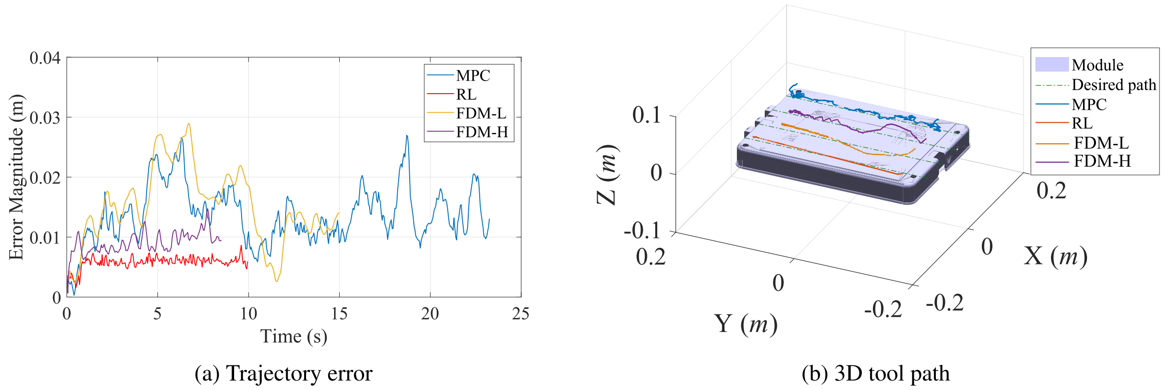
The results for case study 2 are depicted in Figure 4, where it can be observed that the tool requires approximately 23 seconds to reach the endpoint when using MPC, 15 seconds with FDM-L, approximately 9 seconds with FDM-H and the TD3 agent completes the task in 10 seconds. The performance of the RL agent for case study 2 can similarly be compared by analyzing the 3D TCP path, as shown in Figure 4b. Despite all approaches attempting to correct any deviation from the desired path, the deviation is again less prominent in the RL method. This finding is further supported by the RMSE tracking error, where the FDM-L method exhibits the highest RMSE value of 11.1mm, followed by MPC with an RMSE of 10.4mm. In contrast, the FDM-H method achieves a lower RMSE of 4.5mm, while the RL method stands out with a notably lower RMSE value of 0.84mm.



(a) Trajectory error

(b) 3D tool path

**FIGURE 3.** Case study 1: material stiffness $k_p = 9.22116 \times 10^7 (N/m)$, friction coefficient $\mu = 0.95413$.



(a) Trajectory error

(b) 3D tool path

**FIGURE 4.** Case study 2: material stiffness $k_p = 2.19674 \times 10^6 (N/m)$, friction coefficient of $\mu = 0.797441$.

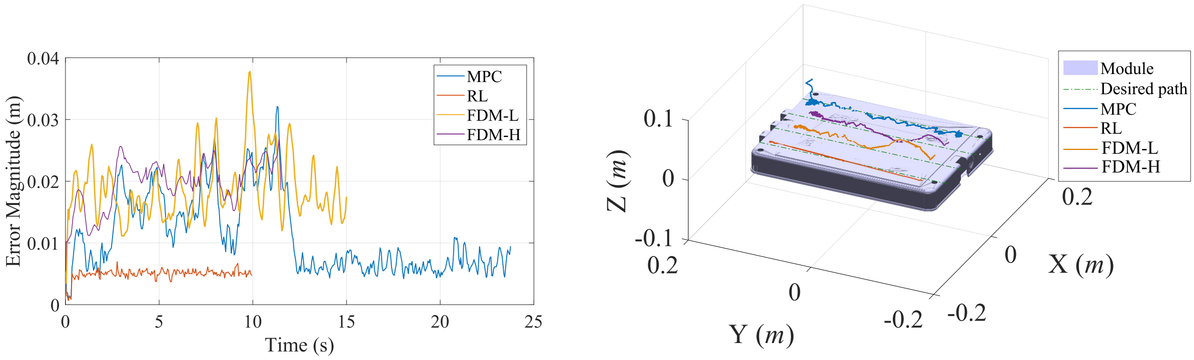
Figure 5 presents the outcomes of case study 3, with the endpoint again reached in roughly 23 seconds when using MPC, 15 seconds using FDM-L method, roughly 12 seconds when using FDM-H and 10 seconds when using the trained RL agent. From Figure 5a it is also evident that the magnitude of the trajectory error with the trained RL agent is lower than that of the others during path following. The 3D path of the tool-tip in Figure 5b provides a basis for comparing the performance of the RL agent with other methods in case study 3. All methods make an effort to correct any deviations from the intended path, but the deviations are less in the RL method. This observation is consistent with the RMSE values, which were significantly lower for RL (0.34mm) compared to the MPC (9.6mm), FDM-L (10.8mm), and FDM-H (9.7mm). Comparing trajectory errors in Figure 6, the RL agent performed 33% faster (10 seconds) than MPC and FDM-L methods (15 seconds). Notably, the RL method was only 1 second slower than the FDM-H method, which completed the task in 9 seconds. It is also illustrated from Figure 6 that the error magnitude during path following using the trained TD3 agent is lower than that of the other methods. Figure 6b similarly demonstrates the desired path is tracked more closely with RL, with a lower RMSE tracking error (0.11mm) compared to MPC (7.6mm), FDM-L (9.8mm), and FDM-H (7.9mm).

**TABLE 5.** Comparing the average parameters for various case studies across different methods.

| Methods | Computational Complexity (s) | Task Completion Time (s) | RMSE (mm) |
|---|---|---|---|
| MPC | $1.6995 \times 10^{-2}$ | 21.5 | 8.70 |
| RL | $6.2841 \times 10^{-4}$ | 10 | 0.46 |
| FDM-L | $3.1411 \times 10^{-3}$ | 15 | 10.32 |
| FDM-H | $2.8987 \times 10^{-3}$ | 9 | 7.78 |

In the case study 5, Figure 7b, the desired path is tracked more accurately with RL once again, with a lower RMSE tracking error (0.67mm) compared to MPC (10.83mm), FDM-L (8.41mm), and FDM-H (6.89mm). Finally, in case study 6 (Figure 8b), MPC fails to complete the task, RL exhibits the lowest RMSE (0.42mm), followed by FDM-L (6.53mm) and FDM-H (12.26mm). The completion times and error magnitudes are also illustrated in Figures 7 and 8 for case studies 5 and 6, respectively.

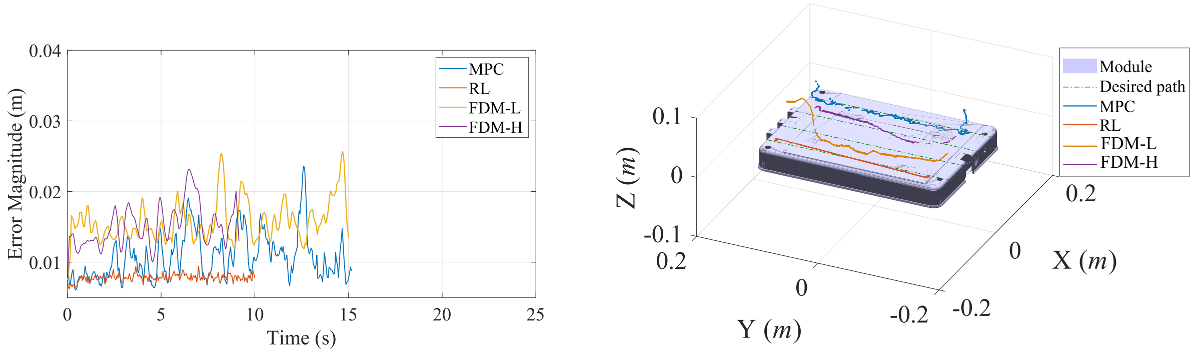
Table 5 provides a summary of results obtained from different approaches to evaluate the average of computational complexity, task completion time, and RMSE between the travelled and desired path. The computational complexity represents the average amount of time it takes for each method to take the observations and return the action for



(a) Trajectory error

(b) 3D tool path

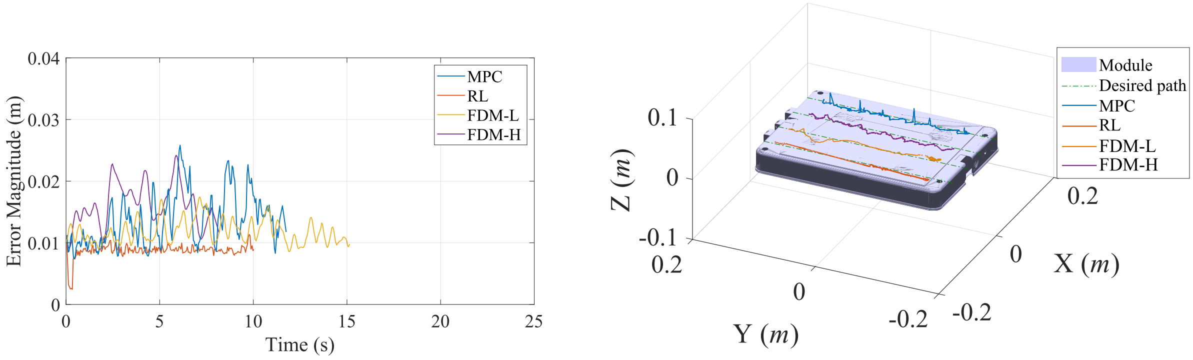
**FIGURE 5.** **Case study 3: material stiffness $k_p = 5.06996 \times 10^5 (N/m)$, friction coefficient $\mu = 1.32379$.**



(a) Trajectory error

(b) 3D tool path

**FIGURE 6.** **Case study 4: material stiffness $k_p = 9.79874 \times 10^8 (N/m)$, friction coefficient of $\mu = 1.40105$.**



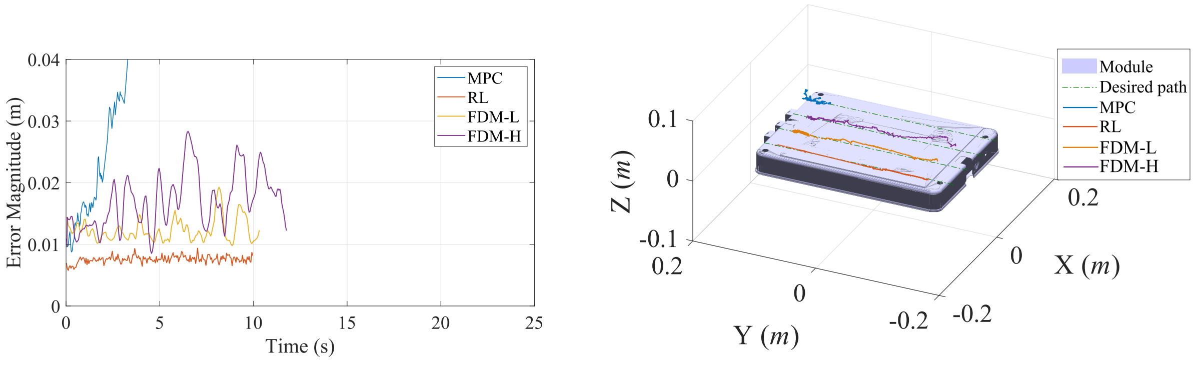
(a) Trajectory error

(b) 3D tool path

**FIGURE 7.** **Case study 5: material stiffness $k_p = 5 \times 10^3 (N/m)$, friction coefficient of $\mu = 0.66567$.**

one time step. From this Table, it is obvious that RL shows the lowest average computational complexity, followed by FDM-H, FDM-L, and finally MPC. RL and FDM-H show the shortest average task completion times, and MPC takes the longest time to complete the task. In terms of accuracy, RL performs the best, with the lowest average RMSE, meaning it stays closest to the desired path. However, FDM-L and MPC have higher values and they deviate more from the desired path. Not to mention that MPC failed to follow the desired path in case study 6.

In summary, the results of the six case studies in Figure 3–7, demonstrate that although all methods are capable of accomplishing the task without prior knowledge of the surface properties, the RL agent outperforms other methods in terms of both speed and effectiveness. This provides evidence for the superiority of the RL agent over MPC and FDM,



(a) Trajectory error

(b) 3D tool path

**FIGURE 8.** **Case study 6: material stiffness $k_p = 2 \times 10^{10} (N/m)$, friction coefficient of $\mu = 0.41343$.**

indicating that the use of RL in completing similar tasks may lead to significant improvements in performance. Moreover, a primary issue with the MPC-based framework is the high computational complexity of the LBMPC method, and hence only approximate solutions for the optimal trajectory may be found at each time step. The degree of variability this introduces leads to the controller exploring areas of the action space for which there is lower confidence in the model predictions, as noted in [6], as they are conditioned on the training data obtained by pre-defined interactions with the system, which are collected offline. Furthermore, LBMPC performance is degraded when extrapolating to new materials outside of the training data. We posit this is due to the accumulation of model prediction error during the inference process of LBMPC, resulting in states diverging from the training dataset, resulting in suboptimal actions being taken. This is most apparent for the high stiffness material in case study 6. In addition, the FDM approach is susceptible to difficulties when generalising to different materials related to choosing suitable stiffness and damping gains, and exhibits a high sensitivity to these parameter values.

## V. TASK 2: EXTENSION TO UNKNOWN, NON-PLANAR SURFACES

We extend the first presented task, considered for the case of unknown material properties with known surface geometry to the more general case where material properties and surface position are both unknown. We procedurally generate heightfield surfaces alongside randomizing stiffness and friction, while modulating the contact force to avoid damage to the tool or workpiece. In doing so, we establish the capability of the trained policy to generalize to various types of surfaces besides the presented planar surface case studies. While in the first instance, the TCP orientation $\boldsymbol{R}$ was excluded, for the case of an unknown, non-planar environment, the rotation encodes useful information about the point of contact and external torques acting on the tool, particularly in the case of loss of visual feedback or occlusion of the surface geometry. We therefore extend the observations from Task 1 to include the TCP orientation $\boldsymbol{R}$, as $\boldsymbol{x} = (d, s, \Delta\boldsymbol{p}, \boldsymbol{f}_e, \boldsymbol{R})$ where the sine and cosine of each Euler angle component was taken as the scaled orientation inputs. Training for the TD3 agent with the curriculum DR and randomized heightmaps was carried out over $2.81 \times 10^5$ seconds for 2000 total episodes. Similarly to Section IV, the hyperparameters for TD3 were established by trial and error. The TD3 hyperparameters and noise information are summarized in Table 7.

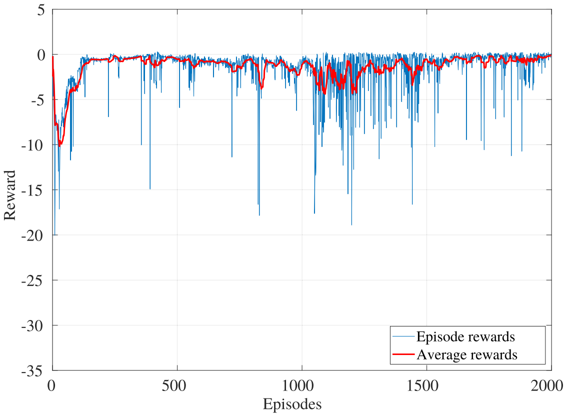
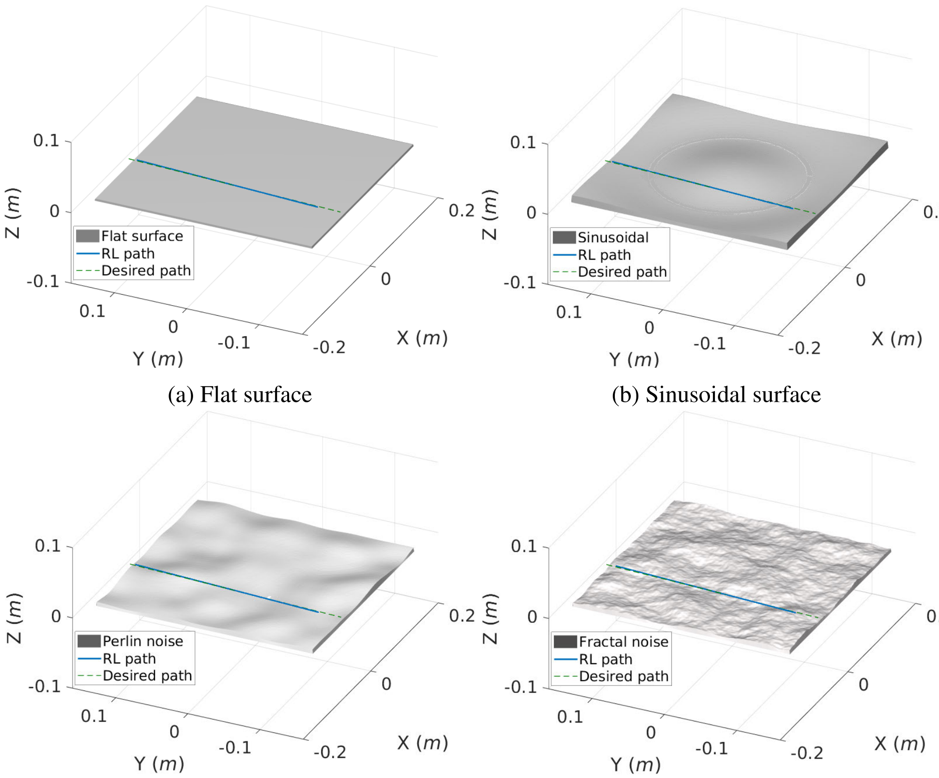


**FIGURE 9.** **Training graph of TD3 agent for compliant surface path following for the case of unknown surface properties and unknown (heightmap) surface geometry.**

The problem of reward function selection is a further necessary and challenging part of task specification. Based on observations in equations (2), (3), and (4), we extend the definition for the agent reward function $r$ as:

$$r = -w_d d^2 - w_s \frac{|s|}{||\boldsymbol{c}||} - w_u \boldsymbol{u}^2 + w_c C - w_f \left(\max\left(f_{max}, ||\boldsymbol{f}_e||\right) - f_{max}\right) \tag{15}$$

where $w_d$, $w_s$, $w_c$ and $w_f$ are manually tuned weighting terms. $w_d d^2$ and $w_s \frac{|s|}{||\boldsymbol{c}||}$ are deviation and slicing terms explained in (10). While these reward contributions alone may be sufficient for unconstrained path following, in the presence of path planning errors presented by an unknown



(a) Flat surface

(b) Sinusoidal surface

(c) surface with perlin noise

(d) surface with fractal noise

**FIGURE 10. 3D perspective view of the cutter TCP path with the curriculum DR trained TD3 agent for different generated height maps.**

**TABLE 6. Reward weighting contributions for the reward function defined in (15).**

| Weighting | Value |
|---|---|
| Path deviation $w_x$ | 1000 |
| Path progress $w_s$ | 2 |
| Effort $w_u$ | 0.3 |
| Contact $w_c$ | 0.75 |
| Force limiting $w_f$ | 0.006 |
| Termination penalty $r_{term}$ | 400 |

**TABLE 7. The reinforcement learning hyperparameters and noise options used in training the TD3 policy.**

| RL parameters | | Noise options | |
|---|---|---|---|
| Smooth factor | 0.001 | Mean | 0 |
| Learning rate | $5\times10^{-4}$ | Mean attraction | 2.5 |
| Sample time ($T_s$) | 0.02 | Variance decay rate | $1\times10^{5}$ |
| Discount factor | 0.99 | Variance | 0.5 |
| Mini batch size | 512 | | |

or approximately known surface geometry, it is necessary to ensure the robot does not apply excessive force to the environment to avoid tool breakage or fail to accomplish the desired tasks by avoiding the surface entirely. This is accomplished by the latter 3 terms. $w_u\boldsymbol{u}^2$ is a small effort penalty to discourage extreme motions. $C$ is a discrete reward contribution encouraging the agent to establish contact with the environment, defined as:

$$C = \begin{cases} 1 & \text{if} f_z > f_{min} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

We introduce a ramping force penalty that penalises forces $(w_f\,(\max\,(f_{max},\,||\boldsymbol{f}||) - f_{max}))$ in excess of a target threshold $f_{max}$. Finally, for training, an additional terminal penalty is applied, defined as $r_{term}$ in the case of early episode termination, and 0 otherwise. Without this penalty, the cumulative reward may converge to a local minimum corresponding to the agent immediately pursuing the episode termination criteria, versus the case of a prolonged episode, where the cumulative penalties due to path deviation or excessive force may be higher. The terminal penalty was chosen to be sufficient to surmount any negative cumulative reward expected during the prolonged episode. The chosen weighting contributions and termination penalty for each reward are shown in Table 6. Weights were adopted as the previous task, and further tuned during preliminary experiments. Notably, the progression term was reduced and effort term increased to discourage saturation of the output velocities and increase the influence of the force and contact penalties. Due to uncertainty in the surface geometry, we found these were comparatively more important to fulfil the task objectives while maintaining surface contact. The values of $f_{max}$ and the contact term were tuned relative to the force limiting term such that a "dead-zone" of force exists during contact. This has the twofold benefit of minimising loss of contact, while also encouraging learning discrimination of contact and non-contact states from observed forces, which improves robustness to uncertainty in the estimated surface positions, or loss of visual feedback entirely. For training, the movements of the tool were furthermore bounded via workspace limitations about the desired path, which we employ in both task space and joint space.

The training was halted when the agent reached the threshold of 2000 episodes, and the training progress graph is shown in Figure 9. Similarly to the training on the planar surface in Section, the performance rapidly converges within the first $\sim$ 250 episodes. For the remainder of training, the reward remains approximately constant, with a progressive degradation of performance from 500–1250 episodes as the range of surface properties is introduced by the curriculum schedule. Finally, the performance recovers for the remaining episodes, indicating successful learning of the task. Figure 10 illustrates the trajectory of the cutter TCP with the trained RL algorithm on different heightmaps. In the case of path planning errors or loss of visual feedback, the reference path may be defined slightly below the object surface. Hence, perfect tracking of the path cannot be achieved without violating the force limiting objectives defined in (15). However, the trajectories of the tool TCP in Figure 10 demonstrate the proposed method results in a learned policy that is robust to an uncertain environment for a variety of surface types.

Due to the robustness of the proposed method to a variety of surface geometries, extension to further applications, for example, path following on deformable objects, such as polishing of flexible surfaces, or cutting of thin, deformable objects remains a compelling avenue for further research. We posit that besides the application presented in this work, practical implementation on a physical robot setup could be carried out with either onboard or wrist-mounted force sensors, in combination with wrist- or externally mounted RGBD cameras. Alternatively, other means of surface position estimation, such as laser distance measurement could be used. However, while the proposed method considers i.i.d depth noise and is robust to loss of visual feedback, other sources of error, such as camera calibration, particularly for the static mounting case could be considered. Furthermore, sensor limitations such as noise, or model mismatch between simulated and real world tasks (for example, extension to surfaces with non-linear stiffness characteristics) remain problematic for real world deployment. Due to the offline nature of data collection, LBMPC remains more readily adaptable to different domains as data can be directly sampled from the target domain. Hence, future work will explore modelling of simulated and real world observations with potentially differing task and sensor dynamics, to enable the proposed method to be adapted across various domains.

## VI. CONCLUSION

In this work, we proposed a TD3 agent with curriculum-based DR to learn contact-rich path following with parametric uncertainties in the interaction contact dynamics. We specifically considered the case of robotic path following along a workpiece with unknown stiffness and isotropic friction over a range of values. By validating our approach with six random case studies in simulation, we demonstrated the robustness of the learned policy representation to unknown environments. Comparison with an earlier approach using LBMPC and a virtual forward dynamic model (FDM), illustrates RL superior task performance with improvement in tracking error; the LBMPC approach suffering due to computational complexity and the problem of adequate domain coverage in the training dataset when employing established expert policies for data collection. Furthermore, the FDM approach is vulnerable to challenges associated with selecting appropriate stiffness and damping gains and is highly sensitive to these parameter values. We extend this concept by procedurally generating heightfield surfaces alongside randomizing stiffness and friction, during the online training, which allowed us to generalize the trained policy for various types of surfaces beyond planar surfaces to environments with unknown surface geometries and path-planning errors.

A notable limitation of the current work is sensor limitations, particularly with loss of visual / depth feedback with reflective or occluded surfaces, or close to the target surface. Although the method compensates for vision feedback loss by incorporating both visual and tactile modalities for path following, the so-called "reality gap" between simulation remains a challenge. To directly bridge this gap, RL methods rely on exploration of the target environment, which is costive on a real setup. Therefore, future work will focus on addressing domain adaptation of the proposed method to a range of target domains, including real world applications. Future endeavours could also benefit from integrating systematic hyperparameter optimization techniques such as Bayesian optimization or Genetic algorithms, instead of the trial-and-error approach used in this paper.

## ACKNOWLEDGMENT

*(Ali Aflakian and Alireza Rastegarpanah are co-first authors.)*

## DATA AVAILABILITY

The implementation of the reinforcement learning algorithms discussed in this paper can be found in this repository: https://github.com/aaflakiyan/RL-Contact-Rich-Path-following.git

## REFERENCES

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.

[2] M. G. Forbes, R. S. Patwardhan, H. Hamadah, and R. B. Gopaluni, "Model predictive control in industry: Challenges and opportunities," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 531–538, 2015.

[3] M. Zanon and S. Gros, "Safe reinforcement learning using robust MPC," *IEEE Trans. Autom. Control*, vol. 66, no. 8, pp. 3638–3652, Aug. 2021.

[4] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, Sep. 2013.

[5] M. Omer, R. Ahmed, B. Rosman, and S. F. Babikir, "Model predictive-actor critic reinforcement learning for dexterous manipulation," in *Proc. Int. Conf. Comput., Control, Electr., Electron. Eng. (ICCCEEE)*, Feb. 2021, pp. 1–6.

[6] G. Bellegarda and K. Byl, "An online training method for augmenting MPC with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5453–5459.

[7] K. Zhang, J. Wang, X. Xin, X. Li, C. Sun, J. Huang, and W. Kong, "A survey on learning-based model predictive control: Toward path tracking control of mobile platforms," *Appl. Sci.*, vol. 12, no. 4, p. 1995, Feb. 2022.

[8] C. Shin, P. W. Ferguson, S. A. Pedram, J. Ma, E. P. Dutson, and J. Rosen, "Autonomous tissue manipulation via surgical robot using learning based model predictive control," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 3875–3881.

[9] A. Rastegarpanah, J. Hathaway, and R. Stolkin, "Vision-guided MPC for robotic path following using learned memory-augmented model," *Frontiers Robot. AI*, vol. 8, Jul. 2021, Art. no. 688275.

[10] A. Padalkar, M. Nieuwenhuisen, S. Schneider, and D. Schulz, "Learning to close the gap: Combining task frame formalism and reinforcement learning for compliant vegetable cutting," in *Proc. 17th Int. Conf. Informat. Control, Autom. Robot.*, 2020, pp. 221–231.

[11] 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. (IROS)*, Nov. 2019, pp. 1010–1017.

[12] X. Zhang, L. Sun, Z. Kuang, and M. Tomizuka, "Learning variable impedance control via inverse reinforcement learning for force-related tasks," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2225–2232, Apr. 2021.

[13] T. Faulwasser, T. Weber, P. Zometa, and R. Findeisen, "Implementation of nonlinear model predictive path-following control for an industrial robot," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1505–1511, Jul. 2017.

[14] J. Matschek, J. Bethge, P. Zometa, and R. Findeisen, "Force feedback and path following using predictive control: Concept and application to a lightweight robot," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9827–9832, Jul. 2017.

[15] L. Meng, S. Yu, H. Chang, R. Findeisen, and H. Chen, "Path following and terminal force control of robotic manipulators," in *Proc. IEEE 16th Int. Conf. Control Autom. (ICCA)*, Oct. 2020, pp. 1482–1487.

[16] A. Maldonado-Ramirez, R. Rios-Cabrera, and I. Lopez-Juarez, "A visual path-following learning approach for industrial robots using DRL," *Robot. Comput.-Integr. Manuf.*, vol. 71, Oct. 2021, Art. no. 102130.

[17] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3803–3810.

[18] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving Rubik's cube with a robot hand," 2019, *arXiv:1910.07113*.

[19] S. Scherzinger, A. Roennau, and R. Dillmann, "Inverse kinematics with forward dynamics solvers for sampled motion tracking," in *Proc. 19th Int. Conf. Adv. Robot. (ICAR)*, Dec. 2019, pp. 681–687.

[20] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1587–1596.

**ALI AFLAKIAN** received the B.Sc. degree in mechanical engineering from Isfahan University of Technology (IUT), Iran, and the M.Sc. degree in mechatronics engineering from University of Tehran, Iran. He is currently pursuing the Ph.D. degree in robotics with the University of Birmingham, U.K. He is currently working on a project called "Reuse and Recycling of Lithium-Ion Batteries" (RELIB), with a focus on automating the process of disassembly of Lithium-Ion batteries using advanced robotics and AI techniques. His research interests include robotics, machine learning, computer vision, and human–robot interaction. He is aspiring to use the most state-of-the-art artificial intelligence approaches in the tasks done by robots.

**JAMIE HATHAWAY** received the M.Eng. degree in nuclear engineering from the University of Birmingham, in 2020, where he is currently pursuing the Ph.D. degree with the Faraday Institution, Reuse and Recycling of Lithium-Ion Batteries (RELIB) Project. His current work focuses on learning and demonstration-based methods to develop generalised interaction control strategies for robotising the process of battery pack disassembly. His research interests include data-driven methods for modelling and intelligent robotic control and their application to contact-rich tasks.

**RUSTAM STOLKIN** (Member, IEEE) is currently the Chairperson of the Expert Group on Robotic and Remote Systems, the Chair of robotics, a fellow of Royal Society Industry, a Professor of robotics, and the Head of the Extreme Robotics Laboratory. He is also an Interdisciplinary Engineer with diverse research interests, although mainly focuses on robotics. He is well known internationally, with an extensive track record in leading major robotics research programs that are directly linked to industrial challenges and industrial stakeholders. His research interests include vision and sensing, robotic grasping and manipulation, robotic vehicles, human–robot interaction, AI, and machine learning.

**ALIREZA RASTEGARPANAH** received the Ph.D. degree in medical robotics from the University of Birmingham, in 2016. Continuing his research pursuits, he joined with the University College London and later the Faraday Institution. He is currently a Senior Research Fellow. He is internationally recognized and currently serves as a Co-PI of the REBELION Project, which focuses on robotizing the process of testing, disseminating, and sorting EV batteries. His research interests include robotics, machine vision, AI, machine learning, and human–robot interaction.