2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Imitation learning for sim-to-real adaptation of robotic cutting policies based on residual Gaussian process disturbance force model

Jamie Hathaway[1,2,†], Rustam Stolkin[1,2], Alireza Rastegarpanah[1,2,†]

***Abstract*— Robotic cutting, a crucial task in applications such as disassembly and decommissioning, faces challenges due to uncertainties in real-world environments. This paper presents a novel approach to enhance sim-to-real transfer of robotic cutting policies, leveraging a hybrid method integrating Gaussian process (GP) regression to model disturbance forces encountered during cutting tasks. By learning from a limited number of real-world trials, our method captures residual process dynamics, enabling effective adaptation to diverse materials without the need for fine-tuning on physical robots. Key to our approach is the utilisation of imitation learning, where expert actions in the uncorrected simulation are paired with GP-corrected observations. This pairing aligns action distributions between simulated and real-world domains, facilitating robust policy transfer. We illustrate the efficacy of our method through real world cutting trials in autonomously adapting to diverse material properties; our method surpasses re-training, while providing similar benefits to fine-tuning in real-world cutting scenarios. Notably, policies transferred using our approach exhibit enhanced resilience to noise and disturbances, while maintaining fidelity to expert behaviours from the source domain.**

I. Introduction

Contact cutting processes feature extensively in a range of applications, most notably manufacturing. However, robotic disassembly applications often struggle with variable product designs, condition uncertainties, and the absence of critical manufacturer information, such as material knowledge for process parameter selection or CAD models for process planning. Hence, for such applications, it is desirable to rapidly adapt to novel products with minimal prior knowledge about the components or fasteners to be cut. With the aid of simulation environments, learning-based approaches have shown promise in adapting to uncertainties across various applications, albeit mainly to non-destructive tasks [1], [2], [3], though sim-to-real adaptation for destructive tasks like cutting or milling, although possible [4], remains challenging.

In the context of robotic cutting, besides safety issues and limited availability of labelled target domain data, notable challenges in sim-to-real adaptation include sensor limitations such as noise, disturbances [5] and residual unmodelled dynamic effects such as chattering, which are challenging to model without laborious identification. Learning-based approaches suffer poor generalisation when dealing with distributional mismatch between source and target domain observations. Furthermore, direct re-training is problematic due to the problem of catastrophic forgetting, resulting in large reductions in performance [6]. Previous approaches to sim to real adaptation of can be broadly categorised into domain adaptive and transfer learning approaches [7]. Various domain adaptive approaches have been proposed based on classifier and discriminator models [8], and for tool wear classification in milling [9]. To surmount the problem of limited target domain data, generative approaches have also been proposed based on synthesis of target domain data [10], [11]. For generative and discriminator-based approaches, in addition to labelled source domain data, a large amount of *unlabelled* target domain data are necessary to train the discriminators. Other approaches have been also proposed based on translation models [12] and learning of unified feature representations between domains [13], [14], [15]. A common assumption is that the conditional distributions of outputs, such as classifiers, are domain invariant, while differences between the domains arise from differences in the marginal distributions over observations (covariate shift), which is not always the case [16]. In this work, we consider the case that the conditional distributions differ between domains, representing the "conditional shift" case from [16]. Other approaches aim to directly compensate measured disturbances on the real setup, as proposed in [17], based on a combination of observer with a Gaussian process (GP) model of position-based disturbances. More specifically to milling, [5] proposed a 4-inertial disturbance observer approach for identification of milling force models. However, the authors suggest that some level of prior knowledge may be required to obtain initial estimates for accurate parameter identification.

Transfer learning approaches contrast domain adaptive approaches in that they aim to improve performance on a target domain via uni- or bi-directional transfer of knowledge between domains. Recently, sim-to-real transfer approaches have been proposed based on augmentation of simulations with models learned from trajectories collected from the real environment [18], [19]. However, these approaches require a large number of real world samples to train. On the other hand, GPs have been demonstrated in other application areas to be efficient at learning from a small number of samples [20], [21]. Alternative sim-to-real transfer approaches have employed system identification methods to optimise physi-

1 Department of Metallurgy & Materials Science, University of Birmingham, Birmingham, UK, B15 2TT.
2 The Faraday Institution, Quad One, Harwell Science and Innovation Campus, Didcot, UK, OX11 0RA.
† These authors contributed equally to this work.
This work was supported by the project "Research and Development of a Highly Automated and Safe Streamlined Process for Increase Lithium-ion Battery Repurposing and Recycling" (REBELION) under Grant 101104241
The authors would like to acknowledge Abdelaziz Wasfy Shaarawy, Carl Meggs and Christopher Gell respectively for assistance with experimental validation, design of material holder and cutter tool for experiments herein.

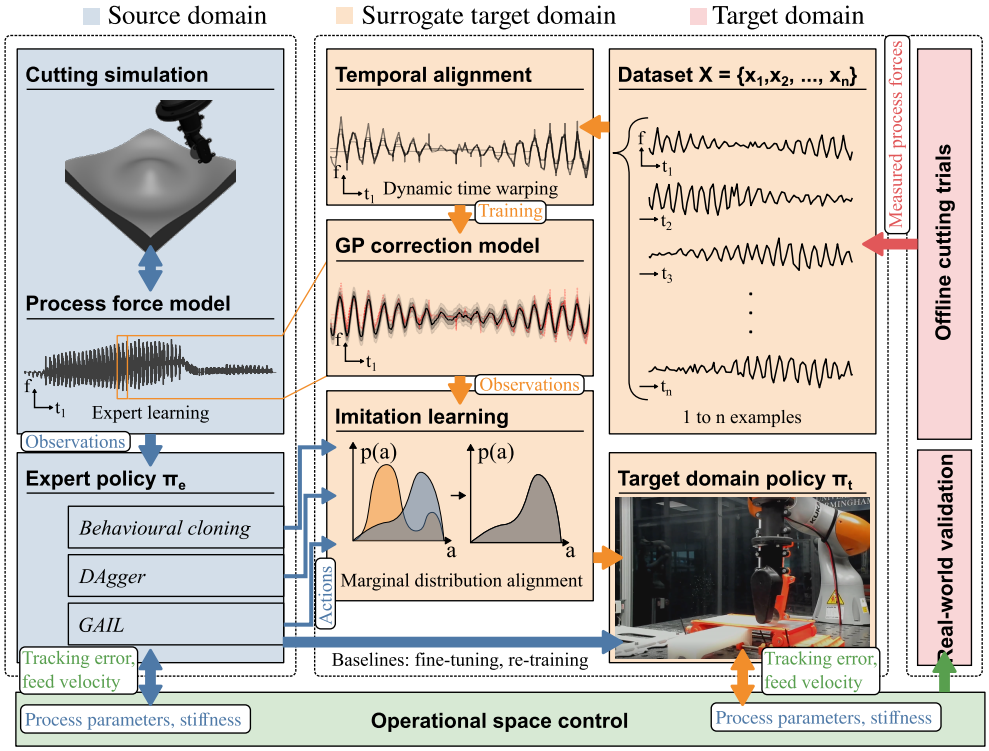


Fig. 1: Overview of the proposed framework, consisting of three stages. In the first stage, a model of the cutting mechanics (source domain) is employed to train an expert policy. Secondly, cutting process force data are collected offline on a target domain (real world), which is used to train a corrective model of disturbances during the real world cutting process. Finally, imitation learning on a surrogate target domain is employed to align the marginal action distributions of expert and learner policies to generate a new target domain policy.

cal parameters of a simulation to maximise target domain performance [22]. Therefore, these methods are capable of only modelling parametric differences between source and target domains. Related to this concept and this present work, [23] leverages an imitation-learning-based approach for domain adaptation based on real world demonstrations on a misspecified simulator.

This paper aims to address challenges in sim-to-real transfer of cutting by learning a GP model of residual process dynamics from minimal (<20) real world demonstrations. To address the problem of changing decision boundaries for actions between domains, we propose an approach based on imitation learning (IL) over corrected simulation observations to align the marginal action distributions between source and target domains. We demonstrate the proposed method outperforms direct application of the expert on the target domain. Our approach provides similar benefits to fine-tuning, however, results in policies that are generally more robust to noise and disturbances and correspond more closely to the source domain expert policy behaviour. An overview of our framework is shown in Figure 1.

II. Methodology

A. Dataset Preparation

We consider an initially unstructured dataset comprising time series $\mathcal{D}_u = \{\mathbf{X}_0 \ldots, \mathbf{X}_n\}$. Each times series contains force measurements $\mathbf{X}_i = \{(t_i, \boldsymbol{f_e})\}$. Many periodic disturbances are parameterised as a function of position instead of time [17]. In the absence of position measurements, the data must first be aligned in the time domain. Each time series is first normalised as:

$$\mathbf{X}_i = \Sigma_{\mathbf{X}_i}^{-1} \left(\mathbf{X}_i - \bar{\mathbf{X}}_i\right) \tag{1}$$

where $\bar{\mathbf{X}}_i$, $\Sigma_{\mathbf{X}_i}$ are the mean and standard deviations of each example time series $\mathbf{X}_i$. The alignment in time domain consists of an initial coarse alignment stage, followed by a fine alignment stage. The coarse alignment is performed by maximising the cross-correlation:

$$t_{\text{delay}} = \underset{t}{\operatorname{argmax}} \left((\mathbf{X}_i * \mathbf{X}_j)\right)(t)) \tag{2}$$

Subsequently, the fine alignment is performed with pairwise dynamic time warping (DTW) between each example trajectory pair $\mathbf{X}_i$, $\mathbf{X}_j$. DTW finds an optimal warping path:

$$(l_n^*, m_n^*) = \underset{(l_n, m_n)}{\operatorname{argmin}} \sum_n \frac{d(\boldsymbol{x}_{l_n}, \boldsymbol{x}_{m_n}) w_n}{\sum_{n'} w_{n'}} \tag{3}$$

with respect to a cumulative weighted distance between features $d(\boldsymbol{x}_l, \boldsymbol{x}_m)$ (here Euclidean distance), where w_n are weighting terms. Adopting the convention from [24], the "symmetric2" step pattern was used. To account for variable length examples, we employ an open-ended DTW approach, which allows for more flexible alignment by permitting variable-length warping paths, which further computes the minimum (3) over all truncated sequences of $\mathbf{X}_j$. To re-index a time series according to the warping path, each element of the original time series is assigned to a new time point based on the optimal mapping. This re-indexed time

series $\hat{\mathbf{X}}$ effectively accounts for time distortions, allowing synchronization and alignment of the data across different examples. Hence, the final dataset comprises aligned time series $\mathcal{D} = \{\hat{\mathbf{X}}_0 \ldots, \hat{\mathbf{X}}_n\}$.

B. GP Force Model

We consider the regression problem:

$$\boldsymbol{f}_e = \boldsymbol{f}(t) + \boldsymbol{d}(t) + \epsilon \tag{4}$$

where $f(t)$ represents the force computed from the underlying process force model, while $\boldsymbol{d}(t)$ represents a disturbance force. The disturbance $\boldsymbol{d}(t)$ is assumed to be periodic and with independent and identically distributed (i.i.d) noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$. We model the disturbance force $\boldsymbol{d}(t)$ as a Gaussian Process (GP) with zero mean and covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$

$$\boldsymbol{d}(\boldsymbol{t}) \sim \mathcal{GP}(\mathbf{0}, k(\boldsymbol{t}, \boldsymbol{t}')) \tag{5}$$

where the distribution of observed and unobserved data is modeled as a joint multivariate Gaussian distribution:

$$\begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{d}_* \end{bmatrix} = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^\mathsf{T} & \mathbf{K}_{**} \end{bmatrix}\right) \tag{6}$$

where $\mathbf{K}$, $\mathbf{K}_*$, $\mathbf{K}_{**}$ are the training, train-test and test covariance matrices respectively. The posterior predictive distribution for the disturbance force $\boldsymbol{d}_*$ given test points $\mathbf{X}_*$ is then

$$p(\boldsymbol{d}_*|\mathcal{D}, \boldsymbol{d}, \mathbf{X}_*) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{7}$$

$$\boldsymbol{\mu} = \mathbf{K}_*^\mathsf{T}\left[\mathbf{K} + \sigma^2\mathbf{I}\right]^{-1}\boldsymbol{d} \tag{8}$$

$$\boldsymbol{\Sigma} = \mathbf{K}_{**} - \mathbf{K}_*^\mathsf{T}\left[\mathbf{K} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{K}_* \tag{9}$$

To capture the periodic nature of the disturbance force, we use the exponential sine covariance function

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(\frac{2\sin^2\left(\frac{\pi d(\boldsymbol{t}, \boldsymbol{t}')}{p}\right)}{l^2}\right) \tag{10}$$

with periodicity p, length scale l and Euclidean distance $d(\boldsymbol{t}, \boldsymbol{t}')$. The kernel hyperparameters for the GP model are estimated directly from the data by minimising the negative (marginal) log likelihood

$$\log p(\boldsymbol{d}|\mathbf{X}) = -\frac{1}{2}\boldsymbol{d}^\mathsf{T}\left[\mathbf{K} + \sigma^2\mathbf{I}\right]^{-1}\boldsymbol{d} - \frac{1}{2}\log\left|\mathbf{K} + \sigma^2\mathbf{I}\right| - \frac{n}{2}\log(2\pi) \tag{11}$$

over the hyperparameter space. To make this procedure more robust, we provide some initial estimates for the noise level based on the sensor noise (measured under static loading in free space) and perform the optimisation from different initialisations.

Based on the assumptions of the mechanistic force modelling approach [25], the cutting force $\boldsymbol{f}$ can be related to each cutting edge $p \in 1 \ldots N_p$ of a fluted cutting tool via mechanistic constants $\boldsymbol{k}_c$, $\boldsymbol{k}_e$:

$$\boldsymbol{f}^{\mathcal{P}} = b_p\boldsymbol{k}_e + b_p\boldsymbol{k}_c h_p \tag{12}$$

where b_p is the thickness of the cutting edge, h_p is the uncut chip thickness, computed from the cutting edge rotation angle θ_p in the tool model reference frame, tool feed rate v and spindle speed ω:

$$h_p = \sin\theta_p \frac{v}{N_p\omega} \tag{13}$$

The total cutting force is computed as the sum over cutting elements weighted by the Boolean vector $\mathbf{G} \in \mathbb{B}^{N_p}$ specifying the engagement of each element k with the workpiece.

$$\boldsymbol{f} = \sum_p^{N_p} G_p \mathbf{R}_{\mathcal{P}} \boldsymbol{f}^{\mathcal{P}} \tag{14}$$

where $\mathbf{R}_{\mathcal{P}}$ is the rotation of cutting edge p about the tool axis to the tool model frame M.

C. Imitation Learning Framework

The cutting task is formulated in the Mujoco simulation environment, into which the model of cutting mechanics from (14) is embedded. The expert reward function r expresses a weighted sum (by weights $Q_\cdot$) of task-specific performance metrics MRV (material removed volume) and cutting time t_{cut}, and feasibility reward shaping comprising path error $\boldsymbol{e}$, process force $\boldsymbol{f}$

$$r = Q_{\text{MRV}} \cdot \text{MRV} - Q_{\text{cut}} t_{\text{cut}} - \boldsymbol{e}\mathbf{Q}_d\boldsymbol{e}^\mathsf{T} - \boldsymbol{f}\mathbf{Q}_f\boldsymbol{f}^\mathsf{T} \tag{15}$$

For the presented cutting task, the actions of the agent considered are $\boldsymbol{a} = \begin{bmatrix}\text{diag}^{-1}(\mathbf{K}_p) & \dot{t}_\Delta & \dot{n_\Delta}\end{bmatrix}^\mathsf{T}$, where $\dot{t}_\Delta$ is related to the commanded feed rate as $t_\Delta = \frac{v}{v_n} - 1$, where v_n is a nominal feed rate, and n_Δ is the commanded depth of cut (DoC - here radial depth of cut). The observation vector was defined as $\boldsymbol{o} = \begin{bmatrix}\dot{\boldsymbol{c}}^\mathsf{T}(t)\dot{\boldsymbol{v}}_{EE} & \boldsymbol{e} & \boldsymbol{v}_{EE} & \boldsymbol{f}_e & t_\Delta & n_\Delta & \text{diag}^{-1}(\mathbf{K}_p)\end{bmatrix}^\mathsf{T}$, where $\boldsymbol{c}(t)$ is the reference cutting path, $\boldsymbol{v}_{EE}$ the end-effector velocity. The reward function weights and hyperparameters were selected as our previous work [4]. Due to the problems of catastrophic forgetting with fine-tuning and high cost of data collection in the real environment, we propose an imitation-learning based approach to train a target policy that can adapt to the target domain via a surrogate target domain. We establish a test case comprising "offline" and "online" IL algorithms; behavioural cloning (BC) and DAgger, which we contrast with the case of the expert policy directly transferred with no fine-tuning, and the expert with fine-tuning directly on the surrogate target domain.

During training, we employ the source domain expert policy π_e to train a surrogate target domain policy $\hat{\pi}_t$. To resemble the fine-tuning case, we initialise the target domain policy as $\hat{\pi}_t = \pi_e$. At each step, the tuple $(\boldsymbol{o}_e, \pi_e(\boldsymbol{o}_e))$ is sampled from the base environment using the data collection policy

$$\pi_d = \beta\pi_e + (1 - \beta)\pi_t \tag{16}$$

where β is the non-expert action probability, which is non-zero for DAgger and zero otherwise. Then, the expert observations $\boldsymbol{o}_e$, actions $\boldsymbol{a}$ $(\boldsymbol{o}_e, \boldsymbol{a})$ are modified by sampling

from the posterior distribution (7) for each point in the trajectory to generate new experiences $(\boldsymbol{o}_t, \boldsymbol{a})$. Subsequently, the surrogate target domain experiences are used to update the target policy as the standard behavioural cloning procedure. However, the method is applicable to other IL algorithms such as GAIL or AIRL. This procedure is summarised in Algorithm 1. We employ the proximal policy optimisation (PPO) algorithm for learning and fine-tuning of policies, although the principles are independent of learning algorithm. Note that differences in action distributions are not considered in this work. Under such a scenario, there is a causal relationship not only between the domain and observations, but also domain and actions. We posit the performance in this case could be improved in a manner similar to [16] via importance reweighting. Additionally, entropy-regularised policies such as those trained by SAC may select actions that are more robust to differing action distributions. For each approach, we conduct training with all strategies for 50 episodes, a learning rate of 1×10^{-3} and batch size of 64. For DAgger, β is varied according to a 0–1 linear schedule for 45 episodes.

Algorithm 1 Imitation-learning sim-to-real transfer

Expert policy, π_e, target policy $\hat{\pi}_t$
Source domain (expert) environment $\mathcal{E}$
Disturbance model $p(\boldsymbol{d}'|\mathcal{D}, \boldsymbol{x}, \boldsymbol{d}, \boldsymbol{x}')$
$\hat{\pi}_t \leftarrow \pi_e$
for $i = 0$ **to** N **do**
  while episode **not** done **do**
    Sample $(\boldsymbol{o}_e, \pi_e(\boldsymbol{o}_e))$ from $\mathcal{E}$
    Sample $\boldsymbol{d}' \sim p(\boldsymbol{d}'|\mathcal{D}, \boldsymbol{x}, \boldsymbol{d}, \boldsymbol{x}')$
    $\boldsymbol{o}_t \leftarrow \boldsymbol{o}_e + \boldsymbol{d}'$
    Append $\mathcal{D}_e$ with $(\boldsymbol{o}_t, \pi_e(\boldsymbol{o}_e))$
    Update $\mathcal{E}$ with $a \sim \hat{\pi}_t(\boldsymbol{o}_e)$
  end while
end for
Train $\hat{\pi}_t$ with $\mathcal{D}_e$ as $\mathsf{BC}, \mathsf{DAgger}, \ldots$

D. Experimental Setup

The real world setup consists of a KUKA LBR *iiwa* R820 14kg collaborative robot equipped with a wrist mounted motorised slitting saw tool. Although the *iiwa* has built-in torque sensing capabilities, we consider the case where a force-torque sensor (FT-AXIA 80) is mounted between the cutter tool and the robot flange which measures the process force. The policy controls the robot via an operational space computed torque tracking controller, with control law:

$$\boldsymbol{\tau} = \mathbf{J}^{\mathsf{T}} \left[\boldsymbol{\Lambda}(\boldsymbol{q}) \left[\mathbf{K}_d(t)\dot{\boldsymbol{e}} + \mathbf{K}_p(t)\boldsymbol{e} \right] + \boldsymbol{\Gamma} + \boldsymbol{\mu} \right] \tag{17}$$

where $\boldsymbol{\tau}$ are the commanded joint torques, $\mathbf{J}$ the robot Jacobian, and $\boldsymbol{\Lambda}$, $\boldsymbol{\Gamma}$, $\boldsymbol{\mu}$ the robot dynamic parameters corresponding to inertia, Coriolis / centrifugal forces and gravitational forces respectively. The policy outputs correspond to the controller position gain $\mathbf{K}_p$ and translational setpoint adjustment, consisting of a feed rate modification and DoC offset from the reference trajectory. The damping gain $\mathbf{K}_d$ is computed according to give a critically damped behaviour for the selected position gain. All quantities are referred to in the base frame W. The experimental setup is shown in Figure 2, depicting the base (W) and model (M) frames employed in the modelling approach. For each strategy, we consider a single cutting pass of a planar material by conventional milling. The reference tool path is defined with respect to the material surface position, with a reference DoC of 0mm – hence, the actual DoC is directly selected via the policy DoC offset. Evaluation was carried out similarly to the simulation as (15), where MRV was estimated by recording the TCP position during each trial with respect to the ground truth material surface position to obtain a DoC, and subsequently MRV by the sum of the swept areas during the task.

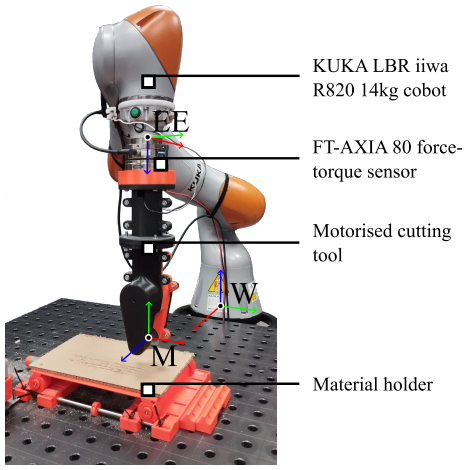


Fig. 2: Overview of the experimental setup used for real world cutting experiments, with tool reference frame $\mathcal{M}$, end-effector $\mathcal{EE}$ and world frame $\mathcal{W}$ shown.

III. Results & Discussion

In this section, the performance of the proposed method is evaluated in simulation in the context of the performance of the expert in the source domain (i.e. simulation without GP augmentation) and surrogate target domain (simulation augmented with GP). The performance and behaviour is subsequently compared for the true target domain for a series of real world cutting tasks.

A. Simulation studies

The GP dataset was constructed from 14 preliminary cutting trials on aluminium and mica, and comprised 35000 samples from 2500-sample windows (5s at 500Hz) per experiment, however, the method is applicable with greater or fewer data. As a proof-of-principle, we consider the size of dataset suitable for offline application on the basis of Algorithm 1. For encapsulating longer-term dependencies, sparse / variational GP approximations could be used to reduce computational complexity. Figure 3 shows the results of temporal alignment of the measurements overlaid with the fitted GP model. Good agreement between the model predictions and the measured forces is shown over the training data which is similarly shown over the transition from training data to extrapolation.

We first establish a case study for cutting of a planar material from a set of 50 trials with randomly chosen mechanistic

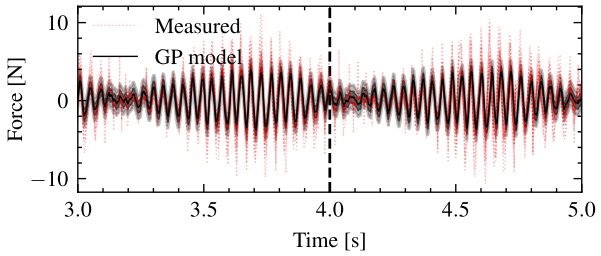


Fig. 3: Plot of measured disturbance force in feed (Y) direction from dataset of examples after temporal alignment. The Gaussian process model fit is shown; the shaded areas show 1-σ and 2-σ standard deviations from the mean respectively. The dashed line shows the transition from training data (left) to extrapolation (right).

constants for the simulation augmented with the learned GP model. We compare the performance of the "expert" agent, which is trained in the unaugmented simulation for 32000 episodes and examine the behaviour with respect to the policies trained with the proposed GP + imitation strategy and fine-tuning, with the performance in the base case, i.e. unaugmented simulation as a benchmark. Figure 4 shows the actions adopted by each strategy, consisting of the relative feed rate versus the nominal (0.75m/min), DoC and controller position gain $\boldsymbol{K}_p$. Correspondingly, the path deviations in the transverse (e_x) and normal (e_z) directions are shown, along with forces in the feed and normal directions (F_y, F_z respectively). Note all quantities are referred to with respect to the robot base frame (W, Figure 2) with uniaxial tool feed antiparallel to the y-axis.

From Figure 4a the behaviour of the expert in the source domain can be seen. When transferred directly into the surrogate target domain ("Expert GP"), the policy reacts aggressively to the added disturbance, exhibiting sporadic variations in feed rate and gain selection. In comparison, the surrogate target domain policies trained with both IL strategies closely tracks the source domain expert behaviour. Conversely, while the actions adopted by the original expert policy after fine-tuning differ, particularly for the Z component of position gain, however, the fine-tuned policy remains similarly sensitive to the disturbances. Comparing the states in Figure 4b further corroborates the similar performance achieved by both imitation learned policies, with DAgger tracking the expert behaviour more closely than BC. The fine-tuned policy adopts a pattern similar to the expert as directly applied to the surrogate target domain, with a delay of ∼0.1s. As an aside, note that with low-pass filtering of the forces, the underlying trend, as seen in the "expert" evaluation, is not recovered, even with aggressive cutoff. This also has the effect of introducing delay into the measured signal; for example, a 10th-order Butterworth filter with a cutoff defined at 5Hz has a maximal group delay of ∼0.5s, or 25 policy evaluations.

Broadly, for all strategies, similar patterns in actions can be observed during each phase of the cutting task. For example, the policy adopts a high position gain and feed rate broadly around the nominal during approach, followed by transitioning to a high feed rate and low gain in the normal direction (Z) after impact. Furthermore, the DoC remains

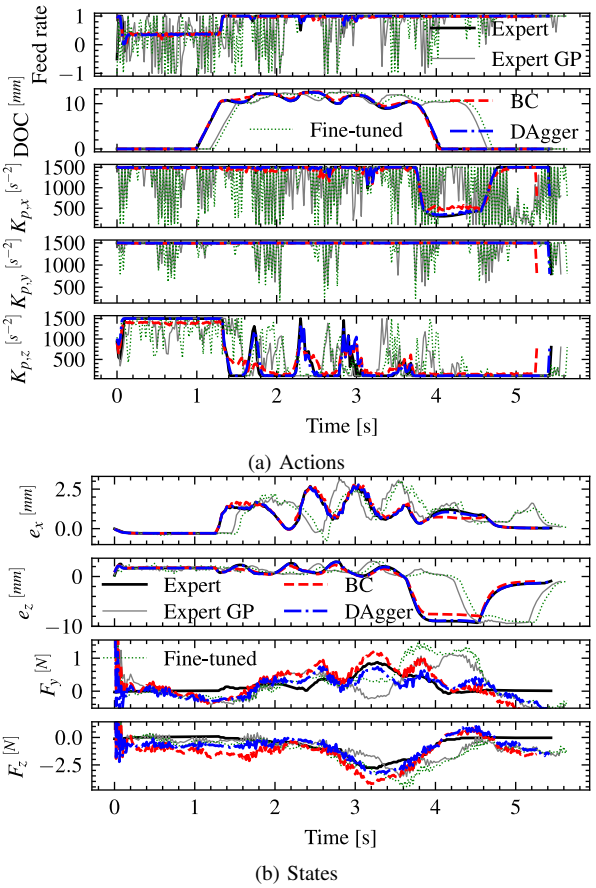


Fig. 4: Comparison of policy actions and states between source domain expert in source and surrogate target domains (Expert, Expert GP), and surrogate target domain policies using fine-tuning, behavioural cloning (BC) and DAgger imitation learning strategies. States include the path error transverse e_x and normal e_z to the planned path and forces in the feed F_y and normal direction F_z. Forces are shown with a 50-point (1s) moving average filter.

consistent across all trials. Therefore, the comparison of each strategy implies predominantly *behavioural* benefits of the proposed approach, rather than fundamentally altering the "decision-making" of the original expert strategy.

Figure 5 shows the training curves for expert policies trained from scratch in source and surrogate target domains respectively. The source policy converges rapidly in the first 3M training steps, before a phase of gradual improvement over the remaining 13M steps to a final reward of -0.766. For the surrogate target policy, an initial reduction in performance is recorded, followed by gradual improvement from steps 5.5–12M until convergence to a final reward of -2.24. Besides the advantage of training time reduction (50 episodes vs. 32000, ∼0.16% of training time), the final rewards from the surrogate target expert policy are notably reduced. This follows from the inclusion of the GP model representing a more challenging task for the agent; the force observations in particular are directly related to the reward and encode important information about the interaction which relates to the selection of milling process parameters.

We consider the performance of the original expert policy, the target domain fine tuned policy and policies with each IL strategy (BC, DAgger) in both the original simulation

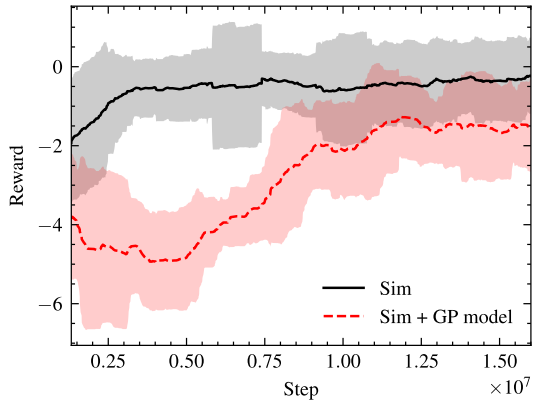


Fig. 5: Comparison of training curves between base environment and environment with GP residual force model showing average rewards with 95% confidence intervals.

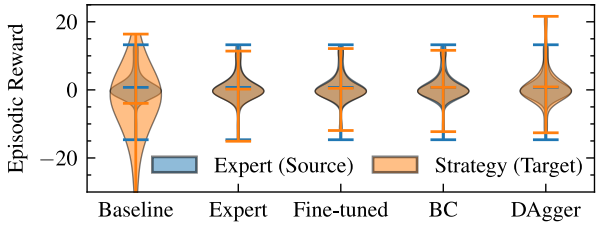
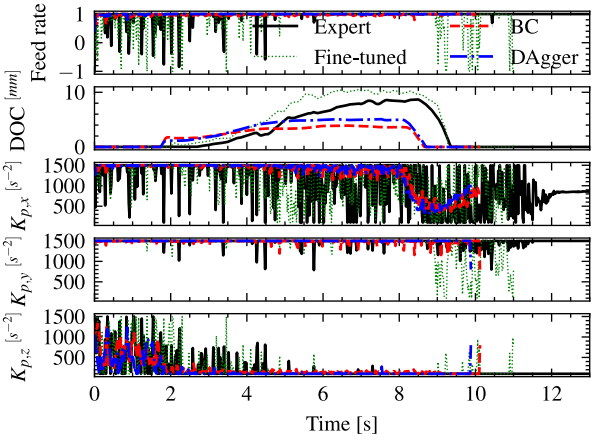


Fig. 6: Violin plot of reward distribution between source domain expert policy and target domain strategies – fixed process parameters for all materials (baseline), unmodified and fine-tuned expert policies, and target policies trained with BC and DAgger imitiation learning approaches.

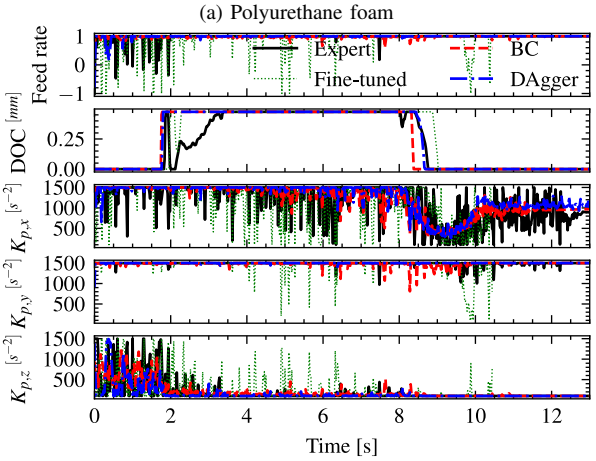
environment and simulation augmented with the learned GP model from real-world trials. We additionally compare the performance with fixed process parameters (Baseline) at the nominal (feed rate 0.75m/min and 1mm DoC) for all materials. In each instance, we compare the episodic rewards over 50 simulation rollouts. From Figure 6 it is clear that the overall performance of the expert and all strategies is robust to distributional mismatch between source and target domains. For the fine-tuned, BC and DAgger strategies, the performance is slightly more consistent in the extreme case as indicated by the higher minimum rewards obtained. The greatest deviation is observed for the DAgger trained target policy from the expert for both the original and GP-augmented environments. As DAgger allows for the trajectories to deviate from those obtained using the expert demonstrations alone, it enables the policy to explore and adapt further to the surrogate target domain. It is therefore unsurprising that DAgger demonstrates superior performance in this case.

B. Real world cutting trials

We evaluate the cutting policy for conventional milling of 5 different materials – high-density polyurethane (PU) foam, corrugated plastic, cardboard, mica, and aluminium – which exhibit broadly differing mechanical and structural properties. Figure 7 and Figure 8 show the actions and key states for the original and fine-tuned experts, BC and DAgger trained policies for foam and mica. Emblematic of the source domain expert as applied directly to the real world is high variability in the agent actions over time, particularly of



(a) Polyurethane foam

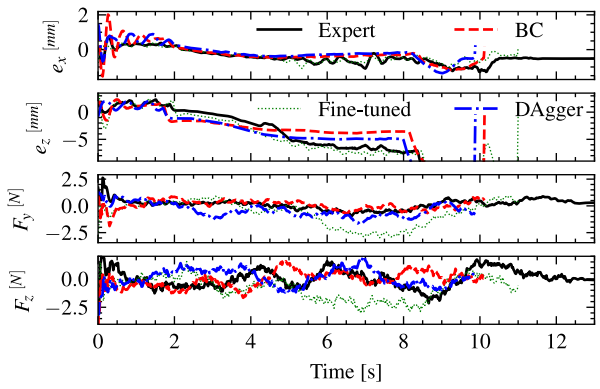


(b) Mica sheet

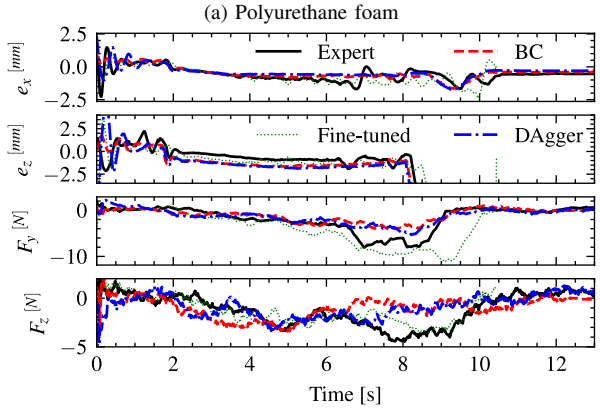
Fig. 7: Comparison of policy actions between source domain expert and surrogate target domain policies using behavioural cloning (BC) and DAgger imitation learning strategies. Actions include the relative feed rate adjustment vs. nominal (0.75m/min), depth of cut (DoC) and controller position gain $\mathbf{K}_p$.

the controller position gain $\mathbf{K}_p$, corroborating the behaviour observed in simulation. The behaviour of the expert and fine-tuned policies are inconsistent, with path error, DoC and force progressively increasing during the cutting task for foam, whereas for the BC and DAgger policies, these are relatively consistent. For the mica cutting task, this is further reflected in the DoC, which is improved for the fine-tuned policy. Although all policies saturated at a similar DoC, the expert and fine-tuned policies were inferior at regulating the process force, which increased in the feed direction towards the end of the task (7-9s) for the expert, and more extremely for the fine-tuned policy. Note that since the focus of this work is on "rough" cutting for disassembly or decommissioning applications, and not on achieving high dimensional tolerances, the expected errors are higher than would be expected for milling, e.g. for a manufacturing application.

Some behavioural characteristics are not preserved from the simulation case study; for example, the selected feed rate during the approach stage of cutting remains nearly at the maximum for all strategies, contrasting the with simulation case. Similarly, the position gain along the normal direction (Z) is increased during the approach phase, but to a lesser



(a) Polyurethane foam



(b) Mica sheet

Fig. 8: Comparison of path error transverse e_x and normal e_z to the planned path and forces in the feed F_y and normal direction F_z between source domain expert, BC and DAgger surrogate target domain policies. Forces are shown with a 50-point (1s) moving average filter.

extent than in simulation. However, during the cutting phase, these are broadly more similar to the simulation case. As previously established, the force observations encode key information about the interaction, including the discrimination of contact and non-contact states.

The performance metrics - process time, average path deviation, average tool load and MRV - of the expert, fine-tuned, BC and DAgger trained target policies are compared over all materials, with five cutting trials per material in Figure 10, while the reward components and average reward for each material is shown in Figure 9. Comparing the expert with direct fine-tuning and with re-training on the surrogate target domain, the fine-tuned policy had consistently lower path deviation and higher MRV, but greater force than the expert in all cases. The re-trained policy similarly had higher MRV, albeit much greater path deviation and slower task execution and for all materials except aluminium, higher process force than the baseline, making its overall performance inferior to the other policy-based strategies, corroborating the training process in Figure 5 and emphasising the importance of retaining source domain knowledge for the cutting application. On the other hand, both policies trained with BC and DAgger had consistently lower process force than the other policy-based strategies across all materials. Similarly, path error was consistently lower than the expert for BC and DAgger, outperforming the fine-tuned policy. Process time

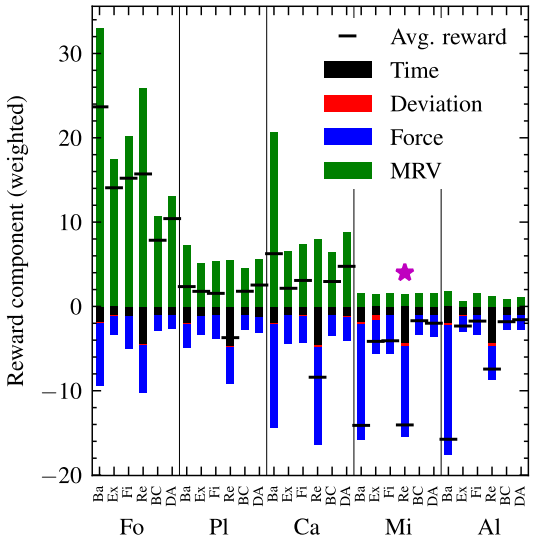


Fig. 9: Breakdown of average reward and reward components by strategy and by material, abbreviated Foam, Plastic, Cardboard, Mica and Aluminium respectively. Strategies marked with a star were unable to complete the task; the reward is estimated from the single 'closest to successful' trial.

was lowest with the BC and DAgger strategies; DAgger performed best on foam, mica and aluminium, while BC performed best on plastic and cardboard, although reductions in time amongst the policy-based strategies were limited (approx. 3–9% relative to expert).

Adopting the unpaired two-tailed T-test (unequal variances) for overall average rewards, the re-trained policy exhibited inferior performance for all materials ($p < 0.001$), whereas fine-tuning yielded no significant ($p < 0.05$) changes from the expert for any material. Between the IL trained policies and fine-tuned policy, significant changes were found only for the foam trials (BC $t(4) = -5.4, p = 0.0034$; DAgger $t(4) = -3.5, p = 0.018$), and mica trials ($t(4) = 13, p = 1.1 \times 10^{-6}$; $t(4) = 11, p = 3.9 \times 10^{-6}$). This implies while fine-tuning and IL both preserve the performance of the expert, IL generally outperformed the former for the more challenging materials. However, fine-tuning is insufficient to address distributional mismatch in source and target domain actions, leading to inconsistent selection of actions and deviation from behaviour as expected in the source domain. Thus, while the path error is reduced, this comes at the expense of regulating the process force, as shown by the inconsistent force profiles in Figure 8 and metrics in Figure 10.

IV. Conclusion

An imitation-learning based approach for sim-to-real transfer of a robotic cutting policy was proposed. We demonstrate how the residual process dynamics and disturbances can be modelled from a small number of real world trials. We validate the proposed method on a real robot setup, demonstrating the policies transferred to the real world in many cases to have significantly improved performance over the expert as directly transferred from simulation. We also demonstrate the proposed method based on imitation learning (IL) outperforms re-training, while performing similarly to

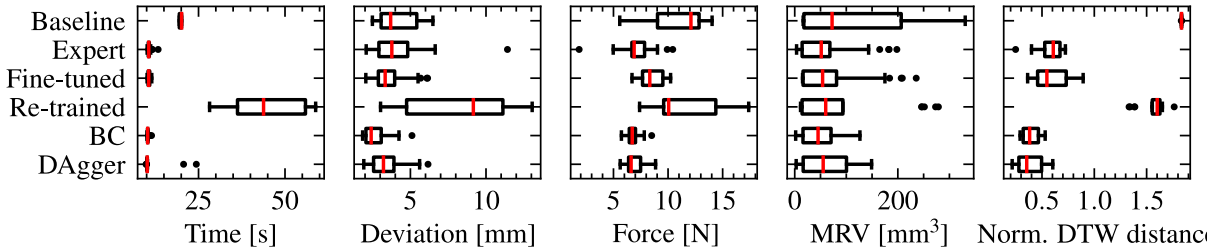


Fig. 10: Distribution of performance metrics for each strategy, aggregated over all materials: completion time, average path deviation, average tool load (lower better) and material removed volume (MRV, higher better). The dynamic time warping (DTW) distance compares the similarity of actions to the expert in simulation over 50 trials; lower values indicate closer behaviour to the expert in simulation.

fine-tuning even with relatively simple offline IL approaches. Furthermore, the behavioural characteristics of the target domain policies (sensitivity to disturbances) were improved relative to these approaches.

A notable limitation of this work is that while the proposed method can incorporate data from multiple examples, the disturbances are assumed to be sampled from the same underlying process. If the disturbances differ between demonstrations, the alignment between demonstrations will be poor. In this case, the proposed framework could be extended through batched or multi-task GP models. The assumption of a periodic disturbance force furthermore remains a notable limitation. To address these limitations, future work will explore direct synthesis of surrogate real-world data from a minimal unstructured dataset of offline demonstrations.

References

[1] Y. Wang, C. C. Beltran-Hernandez, W. Wan, and K. Harada, "Hybrid trajectory and force learning of complex assembly tasks: A combined learning framework," *IEEE Access*, vol. 9, pp. 60 175–60 186, 2021.

[2] X. Li, J. Xiao, W. Zhao, H. Liu, and G. Wang, "Multiple peg-in-hole compliant assembly based on a learning-accelerated deep deterministic policy gradient strategy," *Industrial Robot: the international journal of robotics research and application*, vol. 49, no. 1, pp. 54–64, Jan 2022.

[3] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3803–3810.

[4] J. Hathaway, A. Rastegarpanah, and R. Stolkin, "Learning robotic milling strategies based on passive variable operational space interaction control," *IEEE Transactions on Automation Science and Engineering*, pp. 1–14, 2023.

[5] K. Takahei, N. Suzuki, and E. Shamoto, "Identification of the model parameter for milling process simulation with sensor-integrated disturbance observer," *Precision Engineering*, vol. 78, pp. 146–162, 2022.

[6] S.-I. Ao and H. Fayek, "Continual deep learning for time series modeling," *Sensors*, vol. 23, no. 16, 2023.

[7] E. Salvato, G. Fenu, E. Medvet, and F. A. Pellegrino, "Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning," *IEEE Access*, vol. 9, pp. 153 171–153 187, 2021.

[8] M. Ragab, E. Eldele, W. L. Tan, C.-S. Foo, Z. Chen, M. Wu, C.-K. Kwoh, and X. Li, "Adatime: A benchmarking suite for domain adaptation on time series data," *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 8, May 2023.

[9] K. Li, M. Chen, Y. Lin, Z. Li, X. Jia, and B. Li, "A novel adversarial domain adaptation transfer learning method for tool wear state prediction," *Knowledge-Based Systems*, vol. 254, p. 109537, 2022.

[10] C.-B. Chou and C.-H. Lee, "Generative neural network-based online domain adaptation (GNN-ODA) approach for incomplete target domain data," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–10, 2023.

[11] D. Zhang, W. Fan, J. Lloyd, C. Yang, and N. F. Lepora, "One-shot domain-adaptive imitation learning via progressive learning applied to robotic pouring," *IEEE Transactions on Automation Science and Engineering*, pp. 1–14, 2022.

[12] P. M. Scheikl, E. Tagliabue, B. Gyenes, M. Wagner, D. Dall'Alba, P. Fiorini, and F. Mathis-Ullrich, "Sim-to-real transfer for visual reinforcement learning of deformable object manipulation for robot-assisted surgery," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 560–567, 2023.

[13] Y. Zhang and B. D. Davison, "Deep spherical manifold gaussian kernel for unsupervised domain adaptation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun 2021, pp. 4438–4447.

[14] Y. Zhao, C. Liu, Z. Zhiwei, K. Tang, and D. He, "Reinforcement learning method for machining deformation control based on meta-invariant feature space," *Visual computing for industry, biomedicine, and art*, vol. 5, p. 27, 11 2022.

[15] J. Xing, T. Nagata, K. Chen, X. Zou, E. Neftci, and J. L. Krichmar, "Domain adaptation in reinforcement learning via latent unified state representation," *CoRR*, vol. abs/2102.05714, 2021.

[16] K. Zhang, B. Scholkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *International Conference on Machine Learning*, 2013.

[17] H. Jung and S. Oh, "Gaussian process and disturbance observer based control for disturbance rejection," in *2022 IEEE 17th International Conference on Advanced Motion Control (AMC)*, 2022, pp. 94–99.

[18] F. Golemo, A. A. Taiga, A. Courville, and P.-Y. Oudeyer, "Sim-to-real transfer with neural-augmented robot simulation," in *Proceedings of The 2nd Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Billard, A. Dragan, J. Peters, and J. Morimoto, Eds., vol. 87. PMLR, 29–31 Oct 2018, pp. 817–828.

[19] P. F. Christiano, Z. Shah, I. Mordatch, J. Schneider, T. Blackwell, J. Tobin, P. Abbeel, and W. Zaremba, "Transfer from simulation to real world through learning deep inverse dynamics model," *CoRR*, vol. abs/1610.03518, 2016.

[20] K. Wang, J. Ma, K. L. Man, K. Huang, and X. Huang, "Sim-to-real transfer with domain randomization for maximum power point estimation of photovoltaic systems," in *2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, 2021, pp. 1–4.

[21] S. Eleftheriadis, O. Rudovic, M. P. Deisenroth, and M. Pantic, "Gaussian process domain experts for model adaptation in facial behavior analysis," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Los Alamitos, CA, USA: IEEE Computer Society, Jul 2016, pp. 1469–1477.

[22] M. Kaspar, J. D. Muñoz Osorio, and J. Bock, "Sim2real transfer for reinforcement learning without dynamics randomization," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4383–4388.

[23] S. Jiang, J.-C. Pang, and Y. Yu, "Offline imitation learning with a misspecified simulator," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS'20, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Red Hook, NY, USA, 2020, pp. 8510–8520.

[24] T. Giorgino, "Computing and visualizing dynamic time warping alignments in R: the dtw package," *Journal of statistical Software*, vol. 31, pp. 1–24, 2009.

[25] E. Armarego and R. Brown, *The Machining of Metals*. Prentice-Hall, 1969.