Supplemental Materials

Toward Fine Contact Interactions: Learning to Control Normal Contact Force with Limited Information

Jinda Cui†, Jiawei Xu‡, David Saldana‡, and Jeff Trinkle‡

A. Analysis of the Chiral Problem

In the paper, we introduced the chiral problem, which flips the sign of $\dot{f}_n$ at different contact regions. Without further analysis, we could throw this problem to machine learning. However, it is crucial to understand if the contact region information can be inferred at all. Otherwise, the learning may be destined to failure. Here, we take a closer look into this problem.

What the learned policy regulates is essentially the relative normal speed between the two contacting surfaces. Let us denote the relative normal speed as v_{rel} and the normal velocity offset at the contact as v_{ofs} (note that this is a linear velocity offset resulted from a_{ofs}).

Fig. 1 enumerates all possible combinations of contact regions and relative velocity directions. If the robot knows $|v_{rel}|$, it can always generate a v_{ofs} bigger than $|v_{rel}|$ and check the sign of $\dot{f}_n$. In this case, if v_{ofs} and $\dot{f}_n$ have the same sign, the contact is in one region; if the sign is different, the contact is for sure in the other region. However, v_{rel} is not a measured value (and is difficult to measure). The policy can only observe its immediate previous action and the resulting force change, $\dot{f}_n$, not v_{rel} or $|v_{rel}|$.

Now depending on the relationship between v_{ofs} and $|v_{rel}|$, we can divide the scenarios into three segments for each of the four contact cases. The segments are listed below each diagram. In each table, the first row states the three-scenario segments. The second row reflects the sign of $\dot{f}_n$ resulting from the action. For each segment, the sign of $\dot{f}_n$ is deterministic, and we may use the sign of v_{ofs} and $\dot{f}_n$ as a feature to infer the contact region. The third row is the extracted feature, $sign(\dot{f}_n v_{ofs})$. For the first and third scenario segments, the feature states the contact is in the left region if it is -1 and the right region if 1. This conflicts with the second scenario segment, which states the opposite. Thus, the controller cannot distinguish the contact regions with the inputs used in the examples.

In our implementation, v_{ofs} is not observable. However, the above conclusion still holds if we use a_{ofs} to replace v_{ofs} in the feature, since $sign(v_{ofs}) = sign(a_{ofs})$.

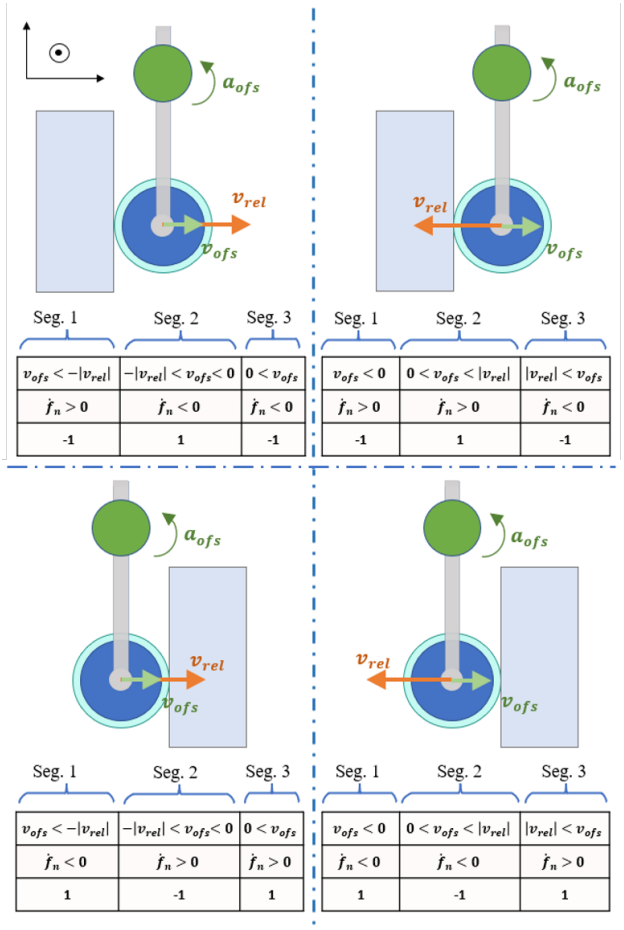

Fig. 1. All scenarios of contact regions and relative velocity directions.

Following the same analysis, we noticed that when the $\ddot{f}_n$ is also given, the contact region can be determined. However, this observation is noisy in simulation, probably due to integration errors, and especially noisy in the hardware robot system when the communication latency and noise come into play. Instead of relying on the $\ddot{f}_n$, an alternative is to let the policy observe a sequence of previous action-response pairs. This would also be useful if the policy learned "probing" actions that may embed information in the sequence.

*This work was partially supported by the National Science Foundation through EFRI C3 SoRo (award 1832795)

‡Autonomous and Intelligent Robotics Laboratory (AIRLab) and the Department of Computer Science and Engineering, Lehigh University, Bethleham, PA, USA. <jix519, das819, jct519>@lehigh.edu

†Honda Research Institute USA, San Jose, CA, USA. (Work was done during the Ph.D. study at Lehigh AIRLab). jinda.cui@gmail.com

B. Additional Results

1) Baseline comparison: Typically, in absence of the chiral problem, normal force magnitude control can be easily achieved with PID control. How would our learned controller perform against a fine-tuned PID controller in such a scenario? Fig. 2 shows typical control plots of the two controllers. When the step-change in target force happens, there can be a delay in response from both controllers, and the rise time following that is non-negligible. Qualitatively, the learned controller's performance is close to the baseline. However, the steady-state performance is not as smooth as the baseline, which has near-zero steady-state errors.

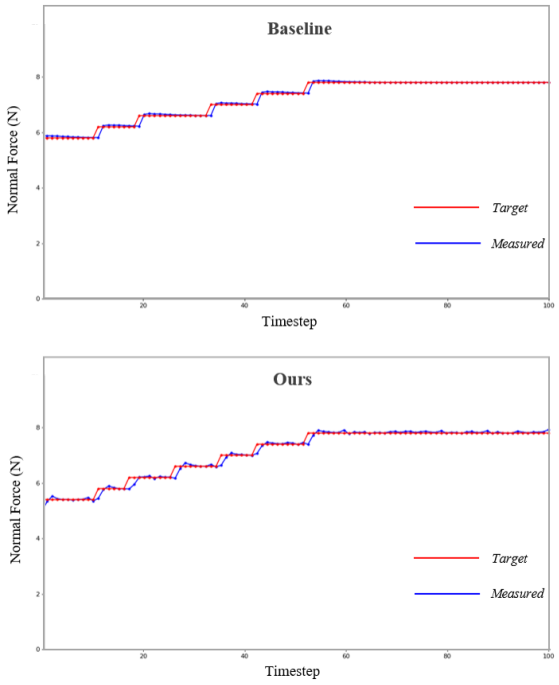

Fig. 2. Typical control measurements during force increments for the baseline and our learned policy.

Nonetheless, in practical scenarios, the chiral problem may present and the relationship between the joint action and the normal force may not be easily determined (*e.g.*, the controlling joint is changing or the object is moving). In such cases the PID baseline would not be able to handle the chiral problem, thus would not be functioning properly. The next evaluation further demonstrates the learned controller's ability to overcome the chiral problem.

2) Quantitative Evaluation: So far, we have demonstrated the feasibility of the learned policy in non-prehensile dexterous manipulation tasks. We provide quantitative evaluations of our learned controller here.

Unlike higher-level manipulation tasks that can be evaluated by counting successes and failures, our controller tracks a continuous target force. Countless aspects can be quantitatively evaluated for a nonlinear controller like ours. A selection based on the application is necessary. To this end, we present two quantitative benchmarks to provide insights on control performance and disturbance tolerance.

The learned policy is a primitive skill on top of which a higher-level manipulation controller can (1) change the target normal force; (2) impose a motion. What our policy does is to try to compensate the unwanted motion in the normal direction and reach the target force value. In the manipulation tasks presented previously, we changed the target normal force by making step changes of $0.4N$. The force controller should respond to this command quickly while keeping the control error small. The first benchmark measures the *Mean Absolute Off-set*, e_T, defined in Section 2.1 of [1]. Specifically, starting from initial contact, we increase the target force by a step change, then measure the mean absolute control error for T timesteps. We performed this benchmark on the hardware robotic system at various force levels across the operating range and at various contact angles that cover both contact regions. Results are shown in the left column of Fig. 3.

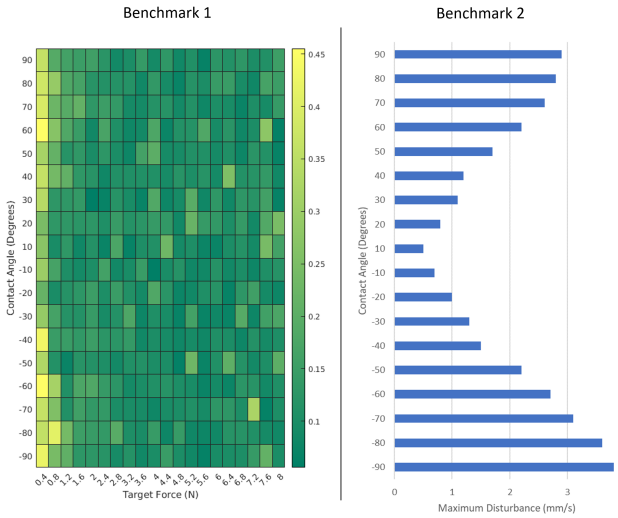

Fig. 3. Left Column: Benchmark 1, $T = 25$. Right Column: Benchmark 2, the initial normal contact force is $5N$.

The overall error is small across the operating range (0.4N-8.0N). However, the performance dropped significantly at a low target force (0.4N in this case), showing room for improvements. Although a low contact angle can be harder to control (since the contact is closer to singularity), the learned policy seems to be unaffected, possibly due to its adaptability.

The second benchmark uncovers the disturbance tolerance (as a ***mode of failure***, when the external disturbance is too much, our controller may fail to recover from it). When a higher-level controller changes the motion, the disturbance it causes can be considered a step velocity change in the normal direction. In the benchmark, starting from an initial static contact, we introduce a step normal velocity change, separating the contact (as approaching could be dangerous to the hardware), and check if the controller can bring the force back to the target (checked for k times, we found $k = 5$ is reliable). If so, the velocity is increased (or decreased if not), and the condition is rechecked. The process is continued

until a small velocity change (0.1 mm/s) flips the result. We performed this test across various contact angles (from -90^o to 90^o). Results are shown in the right column of Fig. 3. Clearly, the performance gets worse when the contact is getting closer to singularity (*i.e.*, 0 degrees). However, at higher contact angles the disturbance tolerance can get up to 3.8 mm/s, which is good enough to enable fine manipulation tasks that are semi-quasi-static.

C. Implementation Considerations

1) Reality gap: The reality gap presents between the simulation and the hardware robotic system. On the hardware, we observed communication latency, measurement noise, and actuation delay. The control frequency is set conservatively to 5Hz as a consequence. In addition, we have to impose a joint speed limit to prevent the system from damaging itself from wild actions.

The key for a successful sim2real transfer is to match the action-force response for each timestep. This can be done by tuning the step response in the simulation. Thanks to the generalizability of learning agents, the tuning do not need to be exact. Our observation is that the transfer can be done as long as the step response is in the same order of magnitude of the real-world response.

2) Domain randomization: The initial contact normal angle are uniformly sampled from $(-90^o, 90^o)$. The joint velocity of the disturbance joint is bounded by one-fourth of the joint action limit. The initial normal contact force is within $[0N, 15N]$

3) Choice of the 0.4N increment: Previous research [2] shows that delta dynamics are much easier to learn for deep learning agents (the agent's actions make small increments rather than dictating the full value). Since our learned force controller is aimed to serve as a low-level controller for higher-level learned manipulation controllers, we focus this study on making small force increments.

REFERENCES

[1] R. Hafner and M. Riedmiller, "Reinforcement learning in feedback control," *Machine Learning*, vol. 84, pp. 137–169, Jul 2011.

[2] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Iterative residual policy: for goal-conditioned dynamic manipulation of deformable objects," 2022.