2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Reinforcement Learning Control of a Reconfigurable Planar Cable Driven Parallel Manipulator*

Adhiti Raman[1], Ameya Salvi[1], Matthias Schmid[1], Venkat Krovi[1]

Abstract— **Cable driven parallel robots (CDPRs) are often challenging to model and to dynamically control due to the inherent flexibility and elasticity of the cables. The additional inclusion of online geometric reconfigurability to a CDPR results in a complex underdetermined system with highly non-linear dynamics. The necessary (numerical) redundancy resolution requires multiple layers of optimization rendering its application computationally prohibitive for real-time control. Here, deep reinforcement learning approaches can offer a model-free framework to overcome these challenges and can provide a real-time capable dynamic control. This study discusses three settings for a model-free DRL implementation in dynamic trajectory tracking: (i) for a standard non-redundant CDPR with a fixed workspace; (ii) in an end-to-end setting with redundancy resolution on a reconfigurable CDPR; and (iii) in a decoupled approach resolving kinematic and actuation redundancies individually.**

I. INTRODUCTION

Cable driven parallel manipulators are parallel robots that employ cables in-lieu of rigid links. The actuation for such a mechanism is provided through cable winching motors that supply tension in the cables to keep the structure rigid. Such robots are lightweight, able to carry large loads in comparison to the required actuation work, can achieve high velocities and can be constructed to navigate over large workspaces.

However, cable driven actuation is accompanied with a set of unique challenges such as unmodeled cable sag/cable dynamics, increased effects of environmental factors (such as vibrations, air resistance), and changing material properties due to tension or wear. In typical CDPRs, several performance indices such as manipulability, stiffness, and accuracy are greatly affected by the pose of the end-effector in the workspace [1] or the geometric configuration of the structure which consequently affects the quality of pose control. Nonetheless, CDPRs can be manually reconfigured to accommodate different workspace reachability requirements and performance qualities by changing geometric attachment points. The value of this reconfigurability was first suggested in a NIST project during the early 2000s [2]. Since then, there have been a number of studies employing reconfigurability to different extents with the goal of improving specific workspace qualities [3], [4], [6].

*This work is supported in part by the U.S. National Science Foundation under NRI grant 1924721

[1]All authors are with the Department of Automotive Engineering at the Clemson University International Center for Automotive Research (CU-ICAR), Greenville, SC 20607 {adhitir,asalvi,schmidm,vkrovi}@clemson.edu

While a number of prior works focus on determining a set of discrete cable attachment configurations (relating to different orientation and quality requirements through optimal design analysis), there is value in exploring an online, real-time reconfiguration controller that can optimize different performance indices at every point in the workspace. However, said controllers are challenging to design, as the redundancy resolution involves computationally expensive non-convex optimizations, and as the resulting dynamics are highly nonlinear.

In this work, we introduce a deep reinforcement learning framework for real-time dynamic control a reconfigurable CDPR (rCDPR) that: (i) can perform optimal trajectory tracking control for a highly redundant dynamic system; (ii) demonstrates the improved overall wrench quality as well as manipulability performance of an rCDPR through the addition of moving cable anchor points; and (iii) improves the computation time to perform this dynamic control (as opposed to classical controllers that require computationally intensive frameworks for online redundancy resolution).

II. RELATED WORK

Limited literature exists that employs kinematic redundancy resolution for parallel robots in an online framework. This is due to the fact that redundancy resolution necessitates the solution of several hierarchical optimization problems at every time step. There is often a need for both local and global optimization schemes, and the computational load increases significantly when there are additional redundancies (such as kinematic redundancies) in the system [5]. In most studies that consider geometric reconfigurability, design constraints are included in the model development to partially or completely eliminate the need for computationally expensive redundancy resolution. For instance, the online reconfiguration planning described by Nyugen et al. [7] utilizes optimization for reconfiguration planning, but bounds the problem by increasing or decreasing the desired workspace along a single dimension. Here, the authors are able to easily define the bounds of the configuration space, thus reducing the nonlinear constraints in the cost function. In other deployments, such as [8], the entire solution space is computed offline and deployed in an open loop for the actual experiment.

Most existing studies limit redundancy planning and control to static and kinematic simplifications of redundant CDPR (rCDPR) models, as task space control and kinematic redundancy resolution schemes are less challenging in this context. Our previous work follows the same principles by

employing a feedforward kinematic controller on a highly dynamic rCDPR model [9]. Dynamic control can already be ambitious in a traditional CDPR model due to the nonlinearity of the system model and cable stiffness [10]. The addition of kinematically redundant joints and associated actuator dynamics intensifies this challenge through additional nonlinearities. For most studies, redundancies are generally resolved using kinematic models compromising dynamic control. Xiong et al. [11] address this challenge by training a network to store a map of the required optimization and by deploying this map in the controller. However, the authors reduce the non-linearity of their cost/reward function by restricting the limits of the movable anchors and workspace boundaries of the end-effector such that they do not overlap. The quality of the wrench feasibility can vary widely depending on the given pose and anchor position, and the maximization of this quality can be locally non-convex. Limiting the anchor and end-effector workspace like this ensures that the anchors are always far away from each other and the end-effector. While this ensures that the non-linear constraints of ensuring wrench feasibility are never violated, it also never explores the theoretical ability of an rCDPR to ensure optimal performance even at the edges of the entire available workspace.

Deep reinforcement learning (DRL) is an attractive solution for complex dynamic control problems as it can be model-free and comprehensive while significantly reducing computation time in deployment. DRL has been successfully applied applications in which control response time is critical [12]. There have been a few successful attempts to incorporate DRL control in CDPRs. Ma et al. [13] successfully apply a Deep Deterministic Policy Gradient (DDPG) framework for tension control on a CDPR by training the network to minimize the distance to a series of randomly generated points. Sancak et al. [14] employ a similar DDPG approach on a planar CDPR while also training for dynamic tracking on randomly generated sine waves. Lu et al. [15] employ a Soft Actor-Critic (SAC) framework to optimize control signals from an inefficient dynamic controller. Additional studies have incorporated reinforcement learning in CDPRs within other contexts, such as identification of parameter biases [16], or vibration suppression [17].

So far, there have been no efforts in expanding the capabilities of DRL to include the control of reconfigurable CDPRs. The additional degrees of freedom for self-motion in combination with the existing static redundancy (in tension determination and wrench feasibility) create a challenging research problem for the application of DRL based control. The ideal redundancy resolution dynamic controller must be able to: (i) complete the trajectory while maintaining wrench feasibility at every point; (ii) maintain the best possible performance index throughout the trajectory; and (iii) account for the system dynamics while maintaining acceptable tracking. In this study, we first develop DRL based trajectory tracking control for a planar CDPR to establish a baseline implementation before expanding to include kinematic redundancies. We apply careful reward shaping arising from system knowledge to arrive at a feasible and satisfying policy. Finally, we demonstrate improved results through the decoupling of the two sets of redundancies and through training them sequentially.

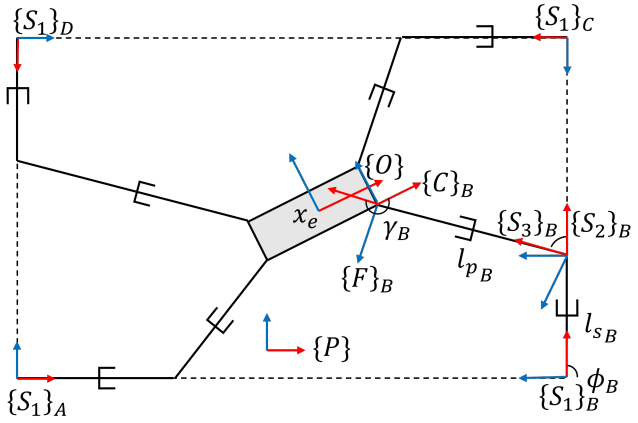

Fig. 1. The reconfigurable 4 cable CDPR

III. CDPR ENVIRONMENT

This section discusses the dynamics of the planar kinematically redundant CDPR that forms the simulation environment for DRL training. The rCDPR is driven by 4 cables whose winches are housed on platforms moved by linear actuators. For the full kinematic model, we refer to our previous work [18]. When the system is stiff and sufficiently tensioned, each cable link forms a prismatic joint between the end-effector. In Fig. 1, $\{F\}_i$ and $\{S_3\}_i$ form the end-effector and base attachment frames of this prismatic cable joint. The joint length is then given by $l_{p,i}$. The tension in the cables is represented by τ_i. The mobile bases, represented by $\{S_2\}_i$ are actuated by linear actuators along the x-axis of the base frames $\{S_1\}_i$. The longitudinal positions of these attachment points with respect to the base frame are given by $l_{s,i}$.

Let $\mathbf{q} = \begin{bmatrix}\boldsymbol{\tau} & \mathbf{l_s}\end{bmatrix}$ be the joint inputs to the plant and realized inputs be $\mathbf{q}^{'} = \begin{bmatrix}\boldsymbol{\tau}^{'} & \mathbf{l}_s^{'}\end{bmatrix}$. The joint sliders are velocity limited to v_{max} and v_{min} through a clamping function i.e.,

$$\mathbf{l}_s^{'k} = \mathbf{l}_s^{'k-1} + \Delta\mathbf{l}_s$$
$$\Delta\mathbf{l}_s = \max(\min(\mathbf{l}_s^{k+1} - \mathbf{l}_s^{'k}, v_{max}\Delta t_d), v_{min}\Delta t_d) \quad (1)$$

where k is the time index of the dynamic simulation, and Δt_d is the timestep of the dynamic system. The full dynamic model of the system can then be derived via a Lagrangian approach as

$$M\ddot{\mathbf{x}}_e + D\dot{\mathbf{x}}_e = P(\mathbf{x}_e, \mathbf{l}_s^{'})\ \boldsymbol{\tau}^{'} \quad (2)$$

where $\mathbf{x}_e = [x, y, \phi]$ is the end-effector pose, and $P(\mathbf{x}_e, \mathbf{l}_s^{'})$ is the pulling map or wrench Jacobian. The mass and damping matrices are M and D respectively. This formulation ignores the effects of gravity (as the system is planar). Finally, the output states of the system are perturbed with zero-mean Gaussian measurement noise.

IV. Metrics for Redundancy Resolution

This section introduces the necessary constraints required for developing the reward function for incorporating redundancy resolution in the RL framework. These are based on established approaches to cost function design used in optimization.

A. Optimal tensions

The tensions in a CDPR must always be positive. The optimal tensions in the tension space are generally selected by either maximizing the stiffness or minimizing the overall tensions in the system. In this study, we choose to incorporate tension minimization in the reward.

$$r = (\boldsymbol{\tau}_{min} - \boldsymbol{\tau})^T (\boldsymbol{\tau}_{min} - \boldsymbol{\tau}) \tag{3}$$

where $\boldsymbol{\tau}_{min}$ is the minimum allowable tension in the cables.

B. Optimal slider positions

The cost function for selecting the optimal slider positions is based on maximizing both the manipulability and the wrench quality of the rCDPR. The best measure for determining the wrench quality in a reconfigurable CDPR is by exploring the Available Wrench Set and maximizing its volume. Determining this value can be computationally expensive. For a CDPR with a one dimensional null space (as in our example), however, a reasonable marker of wrench quality can be determined by calculating the ratio of the smallest and largest tension required to maintain a pose, i.e.

$$\begin{aligned} \mathbf{z} &= \text{null}(P) \\ \kappa &= \begin{cases} \frac{\min(|\mathbf{z}|)}{\max(|\mathbf{z}|)} & \text{if } z_i > 0 \text{ or } z_i < 0 \ \forall \ i \\ -\frac{\min(|\mathbf{z}|)}{\max(|\mathbf{z}|)} & \text{else} \end{cases} \end{aligned} \tag{4}$$

If κ is less than 0, then the system is not wrench feasible. Maintaining a positive value of κ is therefore essential to ensure wrench feasibility. However, good wrench feasibility is not enough, as the system must also be manipulable. In 3 dimensions, the manipulability for translational and rotational motion is considered independently, and this holds true for 2 dimensions as well. Therefore, we only consider the eigenvalues of the twist Jacobian, $P^T(\mathbf{x_e}, \mathbf{l_s})$, corresponding to x and y, $\sigma \in (\sigma_x, \sigma_y)$. The measure of manipulability is then given by,

$$\lambda = \frac{\sigma_{min}}{\sigma_{max}}, 0 \leq \lambda \leq 1 \tag{5}$$

and the manipulability in ϕ is simply given by σ_ϕ. The cable robot can be highly manipulable and still have poor wrench feasibility. If the wrench feasibility is negative, the sign of manipulability is also required to be negative. Since we do not wish to actively control the yaw of the robot, we drop the manipulability in ϕ from consideration.

Maintaining a balance of manipulability and sensitivity ensures that the system moves away from singular configurations without explicitly defining singularity free workspaces to constrain the robot within.

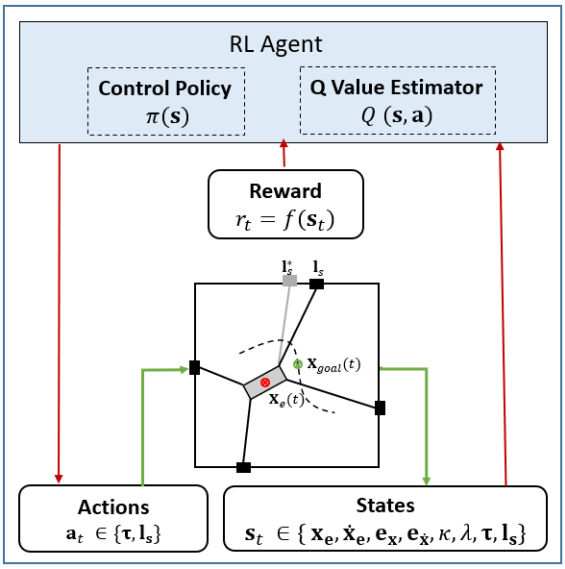

Fig. 2. Overview of reinforcement learning training with a TD3 agent

V. Reinforcement Learning

The goal of DRL control is to select the the optimal control policy that can maximize a cost function. The trained network can be deployed in real-time in lieu of computationally intensive optimization routines applied on cost functions that may not always have an analytical solution.The first step in solving any problem with DRL lies in formulation of the task as a Markov decision process (MDP) [19] with $\mathcal{M} = (\mathbf{s}, \mathbf{a}, \mathbf{p}, r, \gamma)$ being a tuple of the transition probability p, the reward r, states $\mathbf{s}$, actions $\mathbf{a}$, and discount factor γ. Given a state $\mathbf{s}_t \in \mathbf{s}$ and action $\mathbf{a}_t \in \mathbf{a}$ at time t, the probability of transitioning to the next state $\mathbf{s}'_t$ and receiving a reward r_t is given by $\mathbf{p}$. Thus, for a control problem, $\mathbf{p}$ arises from a combination of the dynamics of the system and the reward function defined for training. The goal of the reinforcement learning algorithm, i.e. the agent $\mathcal{A}$, is to learn a control policy $\pi(\mathbf{s})$ through repeated interactions with the environment while maximizing a cumulative reward over a training episode. The success of a DRL control scheme emerges from the meticulous selection of the hyperparameters for the training agent structuring of the control environment and crafting of the reward function. Formulation of the rCDPR control task as a learning problem is discussed in the following section.

A. Problem Formulation

The reference trajectories are structured as Bezier curves through a set of random points selected at the start of every episode during training. A twin delayed deep deterministic policy gradient (TD3) agent [20], an off-policy algorithm advantageous for handling continuous state and action spaces, has been used for training. The agent comprises a total of six neural networks of which one, known as the target actor, is deployed as the control policy $pi(\mathbf{s})$ once the training completes.

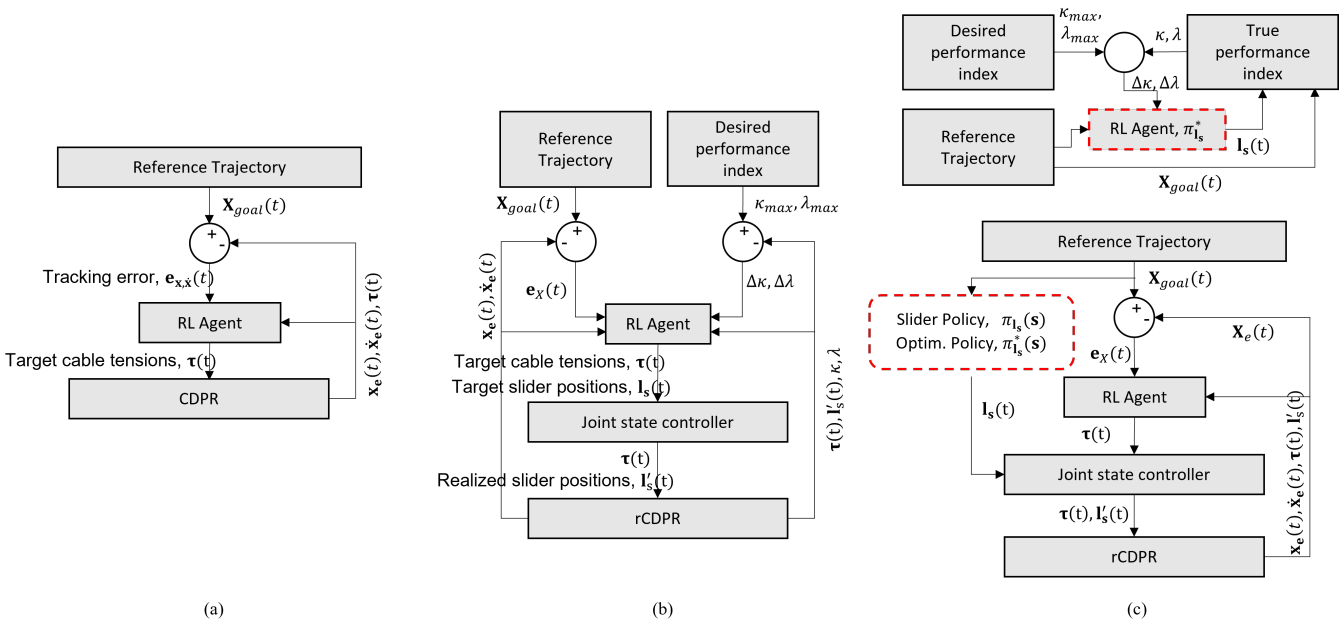

Fig. 3. (a) Tensions only policy for a traditional CDPR (b) End-to-End training for the rCDPR (c) Decoupled policy training for the rCDPR

The actions vector $\mathbf{a}_t$ forms the input to the simulation environment is given by,

$$\mathbf{a}_t = \begin{bmatrix} \mathbf{l}_s & \boldsymbol{\tau} \end{bmatrix} \tag{6}$$

The following states are measured from the simulation and form the state vector $\mathbf{s}_t$,

$$\mathbf{s}_t = \begin{bmatrix} \mathbf{x}_e & \dot{\mathbf{x}}_e & \mathbf{e}_\mathbf{x} & \mathbf{e}_{\dot{\mathbf{x}}} & \kappa & \lambda & \boldsymbol{\tau}' & \mathbf{l}'_s \end{bmatrix} \tag{7}$$

where $\mathbf{e}_\mathbf{x}$ and $\dot{\mathbf{e}}_x$ are the end-effector position and velocity error, κ and λ denote the wrench quality and manipulability index, and $\boldsymbol{\tau}$ and $\mathbf{l}_s$ are the realized tensions and slider positions, respectively.

The reward function $r_t = f(\mathbf{s}_t)$ is calculated from the desired control goal (minimize $\mathbf{e}_\mathbf{x}$ and $\mathbf{e}_{\dot{\mathbf{x}}}$) subject to the constraints discussed in section IV. The episode terminates when the trajectory is completed. The training loop has been illustrated in Fig. 3. The hyperparameters for the different agents are given in Table V-A.

B. Reward shaping

Reward shaping plays a critical role in both the realized speed and quality of learning. The policy behaviour can be tailored by careful crafting of the reward function. Obtained rewards could be designed as sparse (a fixed value of reward achieved for a certain kind of state transition) or continuous (the reward value is a function of state values). Continuous rewards have demonstrated greater success for cases in which knowledge crucial plant dynamics can be encoded in the reward function.

For the rCDPR system there are two primary objectives, trajectory tracking and redundancy resolution. These can be expanded into the following sub-tasks, i.e.

- Position tracking
- Velocity tracking
- Tension minimization
- Wrench quality maximization
- Manipulability maximization

The challenge of manipulating an underdetermined system must be translated into the crafting of the reward function for rCDPR such that the expected behavior can be realized by the learning agent. The final reward function for end-to-end learning then yields

$$\begin{aligned} r = & -w_1||\mathbf{e}_\mathbf{x}||_2 - w_2||\mathbf{e}_{\dot{\mathbf{x}}}||_2 - w_3||\boldsymbol{\tau}_{min} - \boldsymbol{\tau}'||_2^2 \\ & - w_4(1-\kappa) - w_5(1-\lambda) \end{aligned} \tag{8}$$

The first two terms of the reward function emphasize the trajectory tracking performance. The last three terms of the reward function are derived from the constraints in Section IV to optimize the performance indices of the system. Careful selection of the weights, w_i will shape the effectiveness of the training.

TABLE I
HYPERPARAMETERS FOR TRAINING

Hyperparameters	$\mathcal{A}_\tau$	$\mathcal{A}_{\mathbf{l_s},\tau}$	${}^d\mathcal{A}_{\mathbf{l_s}}$	${}^d\mathcal{A}_\tau$
Training Episodes	500	5000	500	1000
Critic Learning Rate	1e-3	1e-3	1e-3	1e-3
Actor Learning Rate	1e-3	1e-3	1e-5	1e-5
Experience Buffer	10000	100000	50000	50000
Batch Size	512	2048	1024	2048

VI. EXPERIMENTS

A. Tensions Only Training, $\mathcal{A}_\tau$

This section forms the implementation baseline for our experiments. In this model, the cable tensions are the only

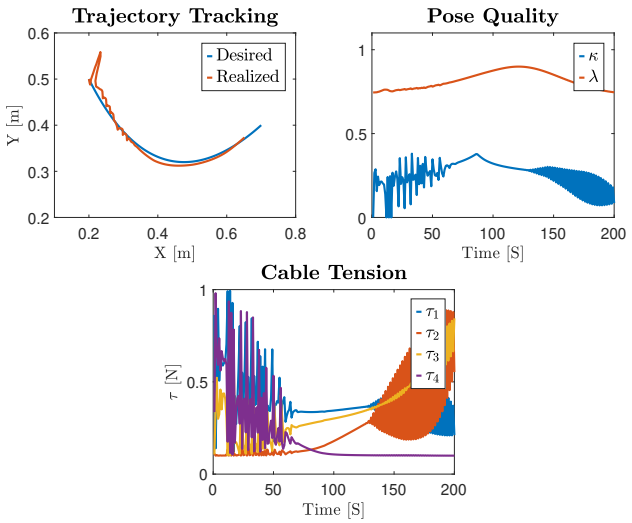

Fig. 4. Tensions only results

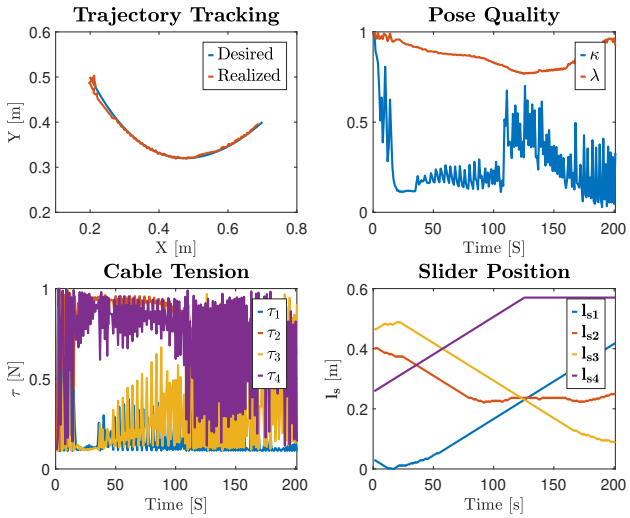

Fig. 5. End to end trajectory tracking

control variable (action), and the sliders are fixed at $l_{s,i} = 0$ (for all i). The reward function used arises as

$$r = -w_1||\mathbf{e_x}||_2 - w_2||\mathbf{e_{\dot{x}}}||_2 - w_3||\boldsymbol{\tau}_{min} - \boldsymbol{\tau}||_2^2 \qquad (9)$$

Here, the third constraint in the reward function minimizes the cable tensions which provides the necessary convexity for selecting the optimal homogeneous solutions. Similar approaches have been discussed in [13], [14] and are therefore not detailed here.

Figure 4 illustrates the operation of this trained agent. The feasible workspace of the system is easily defined, and within this workspace the robot is always manipulable. The wrench quality index is highly dependent on the robot pose so the dynamic fluctuations caused by the perturbations in the actions are reflected in the wrench quality.

It shall be noted that the agent does not control or improve these performance indices as they are solely dependant on the geometry of the structure and are therefore not included in the reward function in Eq. (9). Both values are expected to peak at the center of the workspace and abate towards the edges of the workspace.

B. End-to-end Training, $\mathcal{A}_{\mathbf{l_s},\boldsymbol{\tau}}$

For end-to-end learning, all eight actions $(\mathbf{l_s}, \boldsymbol{\tau})$ were learnt to track a given trajectory and to maximize the performance indices as given by the reward function in Eq. (8). The state vector used in conjunction with the reward function for training is given in Eq. (7). Figure 5 shows that the performance results of the end-to-end approach as implemented were of mixed nature. On the one hand, the implementation did not only demonstrate general feasibility of the learned actions in an end-to-end approach for coupled control with redundancy resolution, but also pose quality exhitibed some overall improvement. On the other hand, the end-effector was subject to oscillations along the trajectory due to large perturbations evident in the tensions. An attempt to reduce these perturbations through training negatively affected tracking quality. The challenges in learning were narrowed down to the size of observation and action spaces and the complexity of balancing multiple constraints in the reward function. Multi-parameter reward functions are difficult to visualize and thus impede the analysis of the combined impact of each parameter on the total reward value. The two input vectors are also unidirectionally coupled, as a change in $\mathbf{l_s}$ also affects the selection of the appropriate cable tensions $\boldsymbol{\tau}$. The effect of the necessary clamping function for the slider positions also introduces a non-linearity which is difficult for the TD3 agent to train for. However, the training results were positively improved by selecting optimal values for the initial slider positions at the start of a training episode (through initial-point numerical optimization).

To reduce the dimensionality of the redundant solution space, the learning task for the two sets of inputs was decoupled in another approach as detailed in the following sections.

C. Decoupled Training, ${}^d\mathcal{A}_{\mathbf{l_s}}$, ${}^d\mathcal{A}_{\boldsymbol{\tau}}$

In this approach, the learning objective was decoupled by training two different agents successively. Here, the first agent ${}^d\mathcal{A}_{l_s}$ controls the slider positions and is trained to learn optimal positions as a function of desired end effector poses. The second agent ${}^d\mathcal{A}_{\tau}$ takes care of the cable tensions that a learned while the policy of the first agent generates slider positions.

1) Training for pose quality: The slider policy ${}^d\pi_{\mathbf{l_s}}(\mathbf{s}_t)$ is learned by incorporating an aspect of imitation learning in the reward function which is in practice similar to the setting of behavior priors for faster training. The priors $\mathbf{l_s^*}$ are determined from numerical optimization of the slider cost function ${}^d\pi^*_{\mathbf{l_s}}(\mathbf{s})$ for a given desired end-effector position. Now, the reward function minimizes the error between action priors $\mathbf{l_s^*}$ and the realized actions $\mathbf{l_s^{'}}$, i.e.

$$r = -w_1||\mathbf{l_s^*} - \mathbf{l_s^{'}}||_2^2 \; - w_2(1-\kappa) \; - w_3(1-\lambda) \qquad (10)$$

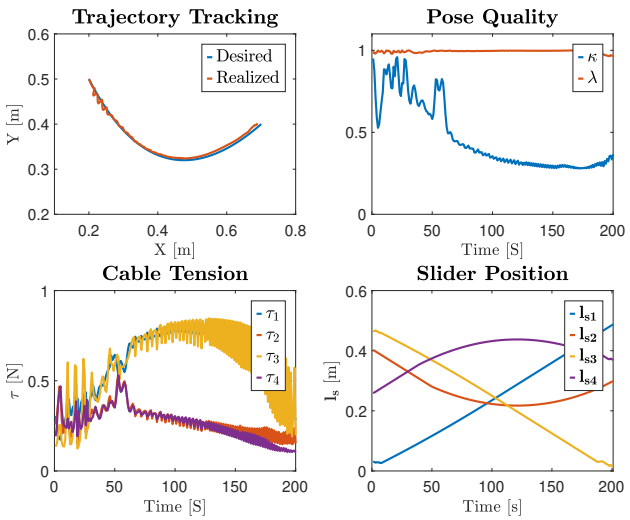

Fig. 6. Decoupled Agent

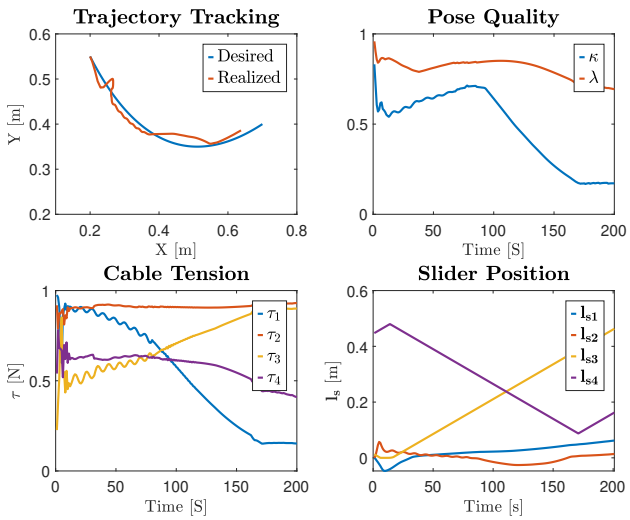

Fig. 7. Decoupled trajectory tracking and Redundancy resolution

Since $\mathbf{l_s^*}$ is computed for the desired trajectory points and supplies discrete discontinuous values at every time step (it does not account for the slider dynamics), the addition of the performance indices to the reward helps optimizing the slider policy for complete trajectories.

2) Training for trajectory tracking: The agent ${}^d\mathcal{A}_\tau$ learns a control policy ${}^d\pi_{\boldsymbol{\tau}}(\mathbf{s}_t)$, for cable tensions using the reward function

$$r = -w_1||\mathbf{e_x}||_2 \ - w_2||\mathbf{e_{\dot{x}}}||_2 \ - w_3||\boldsymbol{\tau}_{min} - \boldsymbol{\tau}'||_2^2 \quad (11)$$

During this training, the slider positions are computed independently. The optimal slider positions can be realized either from optimization ${}^d\pi^*_{\mathbf{l_s}}(\mathbf{s})$ or from the trained policy ${}^d\pi_{\mathbf{l_s}}(\mathbf{s}_t)$ in section VI-C.1:

a) Slider positions from optimization, ${}^d\pi^*_{\mathbf{l_s}}(\boldsymbol{s})$: Here, the tension policy utilizing the optimal slider position $\mathbf{l_s^*}$ was expected to train well, as these slider positions are well behaved and are uniquely reproducible for any given end-effector position in the workspace. This expectation is confirmed in Fig. 6, the tracking is consistent, and the realized slider positions are smooth. However, the wrench quality is not quite optimal since the realized slider velocities are clamped as detailed above in Section III.

b) Slider positions from slider policy ${}^d\pi_{\mathbf{l_s}}(\boldsymbol{s}_t)$: When the trained slider policy is used instead, the overall performance indices are still expected to result in improvement, as the policy incorporates knowledge of the slider dynamics. However, the policy exhibited more complexity in training and shows reduced tracking performance when compared to the previous approach (as seen in Fig. 7). Yet, the wrench quality demonstrates improvement for the same trajectory while manipulability is slightly decreased, but remains on a high level.

Figure 8 illustrates the four difference policies used for inference for hundred random trajectories. The decoupled training strategy trained on the optimization function ($\pi_3 = {}^d\pi_{\boldsymbol{\tau}} \circ^d \pi^*_{\mathbf{l_s}}$) shows the best overall performance.

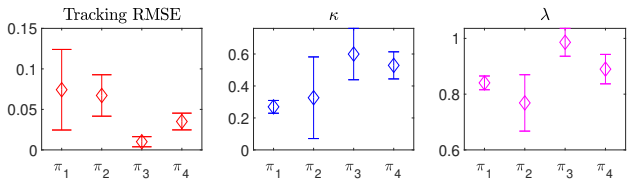

Fig. 8. Mean and SD over 100 random trajectories. $[\pi_1,\ \pi_1,\ \pi_1,\ \pi_1] = [\pi_\tau,\ \pi_{l_s,\tau},\ {}^d\pi_{\boldsymbol{\tau}} \circ^d \pi^*_{\mathbf{l_s}},\ {}^d\pi_{\boldsymbol{\tau}} \circ^d \pi_{\mathbf{l_s}}]$

VII. CONCLUSION

In this work, we analyze different reinforcement learning based approaches to implement end-to-end dynamic control on a reconfigurable/redundant cable driven parallel robot (rCDPR). The results provide an insight into challenges encountered during the training of the off-policy Twin Delayed DDPG (TD3) DRL agent when learning a multi-constraint reward function. Furthermore, the results demonstrate general feasibility while improving performance in comparison with the baseline implementation without redundancy resolution. By decoupling the training of the large and partially coupled input space, the learning process is simplified, and the agents are successfully trained. This is confirmed when comparing the end-to-end and decoupled learning agents directly. Here, the end-to-end agent tries to aggressively jump between multiple local minima to maintain overall optimality of the entire learning episode. The decoupled agents learn by maximizing reduced objectives (trajectory tracking as well as tension minimization and improving pose quality through slider optimization, respectively). Therefore, the previously evident aggressive actions in the cables as well as the resulting fluctuations in the pose quality metrics are reduced. While the overall performance is not yet fully optimal, the decoupled agent shows great promise, and we expect that the trajectory tracking can be improved further with careful reward shaping in future work.

REFERENCES

[1] J.P. Merlet, "Jacobian, manipulability, condition number, and accuracy of parallel robots", *ISRR*, 2005, pp. 199-206.

[2] R. Bostelman, A. Jacoff, F. Proctor,T. Kramer,A. Wavering, "Cable-based reconfigurable machines for large scale manufacturing", *Japan-USA Symposium on Flexible Automation* 2000, pp. 23-26.

[3] L. Gagliardini, S. Caro, M. Gouttefarde, P. Wenger, A. Girin, "A reconfigurable cable-driven parallel robot for sandblasting and painting of large structures" *International Conference on CDPRs*, Springer, Cham, 2015, pp. 275-291.

[4] G. Rosati, D. Zanotto, S.K. Agrawal, "On the design of adaptive cable-driven systems", *Journal of mechanisms and robotics*. 2011, pp. 3(2).

[5] X. Zhou, C.P. Tang, V. Krovi, "Analysis framework for cooperating mobile cable robots", In *IEEE ICRA*, pp. 3128-3133

[6] A. Raman, M. Schmid, V. Krovi, "Stiffness Modulation for a Planar Mobile Cable-Driven Parallel Manipulators via Structural Reconfiguration". *ASME IDETC-CIE*, 2020, Vol. 83990, pp. V010T10A054

[7] D.Q. Nguyen, M. Gouttefarde, O. Company, F. Pierrot,"On the analysis of large-dimension reconfigurable suspended cable-driven parallel robots", *IEEE ICRA*, 2014, pp. 5728-5735.

[8] T. Rasheed, P. Long, A.S. Roos, S. Caro,"Optimization based trajectory planning of mobile cable-driven parallel robots", *In IEEE IROS*, 2019, pp. 6788-6793

[9] A. Raman, I. Walker, M. Schmid, V. Krovi,"A Failure Identification and Recovery Framework for a Planar Reconfigurable Cable Driven Parallel Robots", *IFAC MECC*, 2022, to be published

[10] W. Kraus, *Force control of cable-driven parallel robots*, 2016.

[11] H. Xiong, H. Cao, W. Zeng, J. Huang, X. Diao,W. Lu , Y. Lou, "Real-time Reconfiguration Planning for the Dynamic Control of Reconfigurable Cable-Driven Parallel Robots", *Journal of Mechanisms and Robotics*, 2022, pp. 17:1-57.

[12] A. Salvi, J. Coleman, J. Buzhardt,V. Krovi, P. Tallapragada, "Stabilization of Vertical Motion of a Vehicle on Bumpy Terrain Using Deep Reinforcement Learning", *IFAC MECC*, 2022, to be published

[13] T. Ma, H. Xiong,L. Zhang, X. Diao, "Control of a Cable-Driven Parallel Robot via Deep Reinforcement Learning", In *IEEE ARSO*, 2019,pp. 275-280

[14] C. Sancak, F. Yamac, M. Itik, "Position control of a planar cable-driven parallel robot using reinforcement learning", *Robotica*, 2022, pp. 17:1-8.

[15] Y. Lu, C. Wu, W. Yao, G. Sun, J. Liu, L. Wu, "Deep Reinforcement Learning Control of Fully-Constrained Cable-Driven Parallel Robots", *IEEE TIE*, 2022.

[16] M.M. Aref, J. Mattila, "Automated Calibration of Planar Cable-Driven Parallel Manipulators by Reinforcement Learning in Joint-Space", *In 6th RSI IcRoM*, 2018, pp. 172-177.

[17] H. Sun, X. Tang, J. Wei, "Vibration Suppression for Large-Scale Flexible Structures Using Deep Reinforcement Learning Based on Cable-Driven Parallel Robots", *ASME IMECE*, 2020, Vol. 84546, pp. V07AT07A035.

[18] A. Raman, M. Schmid, V. Krovi, "Wrench Analysis of Kinematically Redundant Planar CDPRs", In *International Conference on CDPRs*, 2021,pp. 90-104.

[19] R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction, Second Edition*, The MIT Press, 2018

[20] S. Fujimoto, H. Herke, D. Meger, "Addressing function approximation error in actor-critic methods." In PLMR ICME, 2018, pp. 1587-1596