Deep Reinforcement Learning for Concentric Tube Robot Control with a Goal-Based Curriculum

Keshav Iyengar and Danail Stoyanov

***Abstract*— Concentric Tube Robots (CTRs), a type of continuum robot, are a collection of concentric, pre-curved tubes composed of super elastic nickel titanium alloy. CTRs can bend and twist from the interactions between neighboring tubes causing the kinematics and therefore control of the end-effector to be very challenging to model. In this paper, we develop a control scheme for a CTR end-effector in Cartesian space with no prior kinematic model using a deep reinforcement learning (DRL) approach with a goal-based curriculum reward strategy. We explore the use of curricula by changing the goal tolerance through training with constant, linear and exponential decay functions. Also, relative and absolute joint representations as a way of improving training convergence are explored. Quantitative comparisons for combinations of curricula and joint representations are performed and the exponential decay relative approach is used for training a robust policy in a noise-induced simulation environment. Compared to a previous DRL approach, our new method reduces training time and employs a more complex simulation environment. We report mean Cartesian errors of 1.29 mm and a success rate of 0.93 with a relative decay curriculum. In path following, we report mean errors of 1.37 mm in a noise-induced path following task. Albeit in simulation, these results indicate the promise of using DRL in model free control of continuum robots and CTRs in particular.**

I. INTRODUCTION

Concentric tube robots (CTRs) are needle-sized robots composed of concentric pre-curved nickel titanium alloy tubes [1]. Individual tubes are super-elastic and have a straight and pre-curved section. By rotating and translating each tube individually, neighbouring tubes interact resulting in bending and twisting of the robot backbone to create curvilinear shapes as seen in Fig. 1. In surgical applications, CTRs are clinically motivated for minimally invasive surgery (MIS) where articulated robots and steerable needles can be used to access surgical sites with minimal trauma. Examples of CTRs in surgical applications include retinal microsurgery [2], endonasal procedures [3], fetal surgery [4], [5] and other procedures [6] all of which may benefit from the dexterity, compliance and flexibility of CTRs. One of the main benefits of CTRs is the small potential profile of the instrument that may minimise trauma at the entry point.

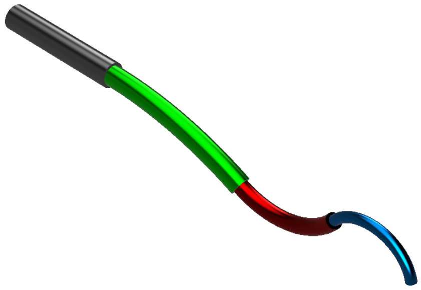


Fig. 1: Curvilinear shape of a three tube CTR.

With the benefits of CTRs, however, comes the inherent complexity of kinematics and control because of the tube interactions that create these curvilinear shapes. Kinematic modelling for CTRs has been investigated thoroughly. A balance between complexity and computation time has resulted in popular kinematic models [7], [8] that are torsionally compliant or in other words, include the twisting of the tubes. These methods have limitations in terms of transverse shear, elongation and friction which are present in real systems. Some relevant works in the field are reviewed in §II. Significant advances have been made such as including friction modelling [9]. However, computing the inverse kinematics (IK) using model-based approaches can be computationally slow, thus a model-free, deep learning approach like deep reinforcement learning (DRL) can include complex phenomena yet be computationally fast and accurate when compared to Jacobian approaches [10].

In this paper, we improve upon a previous model-free reinforcement learning (RL) approach for control of CTRs [11] by improving training convergence time, validating our approach on a more accurate simulation kinematics model and training a robust policy in a noise-induced simulation. These improvements are done by introducing a goal tolerance based curriculum reward strategy, joint representation improvements and training on a noise-induced simulation environment. The contributions of the paper are summarized as:

- Novel training method using a goal-based curriculum reward to learn the control policy for CTRs.
- Train control policy using a geometrically exact kine-

This work was supported by the Wellcome/EPSRC Centre for Interventional and Surgical Sciences (WEISS) at UCL (203145Z/16/Z), EPSRC (EP/P027938/1, EP/R004080/1). Danail Stoyanov is supported by a Royal Academy of Engineering Chair in Emerging Technologies (CiET1819\2\36) and an EPSRC Early Career Research Fellowship (EP/P012841/1).

Keshav Iyengar and Danail Stoyanov are with the Wellcome/ EPSRC Centre for Interventional and Surgical Sciences (WEISS), University College London, London W1W 7EJ, UK keshav.iyengar@ucl.ac.uk

matics model [8], where previous reinforcement work employed a simplified dominant stiffness, constant curvature model derived from [7].

- Train a robust control policy in simulation with added Gaussian noise to demonstrate policy robustness.

II. PRIOR WORK AND PRELIMINARIES

Standard model-based approaches for solving IK and control of CTRs include numerical root finding methods [7], differential kinematics [12] and most recently, a model-predictive controller (MPC) strategy [13]. In the numerical root finding method, a Gauss-Newton root finding is performed in real-time applied on the functional approximation for IK. In the differential kinematics approach, a Jacobian derivative over the arc length of the robot with respect to changes in joints and external forces and torques is derived. The Jacobian is used in a damped least squares differential IK method with an objective function that includes tracking a desired trajectory, stability, actuator velocities, actuator limits and undesirable configurations. Additional work on task space control using an approximate Jacobian has also been investigated [14]. Last, in the MPC approach, a kinematics model was used to predict unstable joint configurations in future joint configurations of the trajectory, and adjust accordingly. A non-linear optimization problem is constructed with a cost function defined over the range of the time horizon K with the current trajectory, desired trajectory and weighting matrix along with constraints to ensure stability. The optimization is solved at each timestep with the horizon set to $K = 5$. Although model-based approaches have made significant progress, in most models, the CTR is seen as a constant curvature pre-curved or straight sections to mitigate the large computation for fast IK. However, during the manufacturing process, this is difficult to ensure. Additionally, the most common material used for concentric tube robots in nickel-titanium, a super-elastic material which will undergo some amount of permanent plastic deformation over time. Lastly, there is a trade-off of computation complexity and model accuracy.

In model-free approaches, there have been four main prior works with three using neural networks and one RL approach. In the first two neural network approaches [15], [16], for forward kinematics, the joint configuration is used as inputs and a fully connected neural network outputs a pose of the end-effector. The inputs and outputs are reversed for IK. Joint configurations are limited to certain areas of the workspace to ensure even data sampling. In the first neural network approach [15], a variable curvature section and constant curvature section three tube CTR was used in simulation. In the second work [16], a novel trigonometric joint representation is used for training which improves accuracy in a real CTR system. Furthermore, various joint representations are compared in the associated work [17]. In the last neural network method [18], the joint configuration represented in the trigonometric form is the input and the outputs are coefficients of a orthonormal polynomial basis function. RGB cameras are used to collect backbone shape images which are voxelised, fitted to the basis function with the coefficients used as training for the neural network. The trained network can then, given a joint configuration, estimate coefficients of the basis functions that best estimate the backbone shape. In the RL approach [11], a policy gradient algorithm is used to train an agent in a simplified piecewise dominant stiffness constant curvature simulation to output a change in joint configurations values given the current state of the robot. The current state of the robot included the trigonometric representation of the joint configuration and the current end-effector position and desired end-effector position. It was found that separating the noise in a multivariate way greatly improved convergence in training. We build on this RL foundational work in CTR control and we improve on it by using a curriculum to decrease training time while using more complex model in simulation. Moreover, a robust policy is trained by using a noise-induced simulation, instrumental for hardware testing.

Generally, RL is a framework that aims to learn a sequence of actions that maximizes a numerical reward to complete a task. The framework consists of an agent or policy, which maps states to actions and an environment, which simulates the selected action in the task space and returns the new state and a reward signal [19]. Generally, there are a number of timesteps with which the agent can interact with environment before it is reset. An episode consists of a number of timesteps and a number of episodes determine the full number of training steps. The aim of RL is to develop a policy that outputs actions which have the the greatest cumulative reward over before completion of an episode. DRL is combining deep learning with RL by using neural networks to represent reward prediction and the learned policy.

III. METHODS

To use RL, a Markov Decision Process (MDP) consisting of state, action and reward must be formulated.

A. MDP Formulation

In the following, the state, action, reward, and goals are defined.

- State (s_t) : States are defined as the concatenation of the trigonometric joint representation, Cartesian goal error and current goal tolerance. As shown in Fig. 2, rotation and extension of tube i (ordered innermost to outermost)

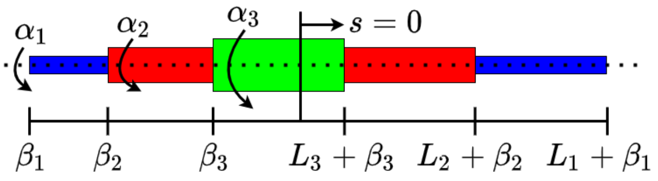


Fig. 2: Joint variables β and α of a 3 tube CTR. s is the arc-length or axis along the backbone.

are α_i and β_i. The trigonometric representation [16] of tube i is defined as:

$$\gamma_i = \{\gamma_{1,i}, \gamma_{2,i}, \gamma_{3,i}\} = \{\cos(\alpha_i), \sin(\alpha_i), \beta_i\} \quad (1)$$

The rotation can be retrieved by taking the arc-tangent

$$\alpha_i = \text{atan2}(\gamma_{2,i}, \gamma_{1,i}) \quad (2)$$

The extension joint β_i can be retrieved directly and has constraints

$$0 \geq \beta_3 \geq \beta_2 \geq \beta_1 \quad (3)$$

$$0 \leq L_3 + \beta_3 \leq L_2 + \beta_2 \leq L_1 + \beta_1 \quad (4)$$

from the actuation side. The Cartesian goal error is the current error of the achieved end-effector position ($G_{achieved}$) and desired end-effector position ($G_{desired}$). Lastly, the current goal tolerance, $\delta(t)$, is included in the state where t is the current timestep t. The full state, s_t, can then be defined as:

$$s_t = \{\gamma_1, \gamma_2, \gamma_3, G_{achieved} - G_{desired}, \delta(t)\} \quad (5)$$

- Action (a_t) : Actions are defined as a change in rotation and extension joint positions.

$$a_t = \{\Delta\beta_1, \Delta\beta_2, \Delta\beta_3, \Delta\alpha_1, \Delta\alpha_2, \Delta\alpha_3\} \quad (6)$$

- Goals (G) : Goals are defined as points in Cartesian space within the workspace of the robot. There is the achieved goal, $G_{achieved}$ and desired goal $G_{desired}$. The achieved goal is determined with the forward kinematics of the model used and is recomputed at each timestep as the joint configuration changes from the agents actions. The desired goal updates at the start of every episode where a desired goal is found by sampling valid joint configurations in the workspace and applying forward kinematics of the model.
- Rewards (r_t) : The reward is a scalar value returned by the environment as feedback for the chosen action by the agent at the current timestep. In this work, sparse rewards are used as they have been shown to be more effective than dense rewards when using hindsight experience replay (HER) [20]. The reward function used in this work is defined as:

$$r_t = \begin{cases} 0 & \text{if } e_t \leq \delta(t) \\ -1 & \text{otherwise} \end{cases} \quad (7)$$

where e_t is the Euclidean distance $||G_{achieved} - G_{desired}||$ and $\delta(t)$ is the goal-based curriculum function that determines the goal tolerance at timestep t. The workspace and various state and reward elements are illustrated in Fig. 3.

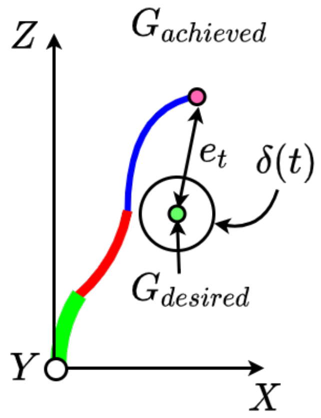


Fig. 3: State with starting goal and achieved goal (red), desired goal (green), goal tolerance, $\delta(t)$. Outer tube (green), middle tube (red) and inner tube (blue).

B. Goal-Based Curriculum

We propose three goal tolerance curriculum functions for training. Using the starting goal tolerance, $\delta(0)$, final goal tolerance $\delta(N_{ts})$, where N_{ts} is the number of timesteps to apply the function, we can fully define our three chosen functions. The first function is a constant tolerance.

$$\delta_{const}(t) = \delta_{const}(0) = \delta_{const}(N_{ts}) \quad (8)$$

The second function is linear function, with b as the initial tolerance and a as the slope.

$$\begin{aligned} \delta_{lin}(t) &= at + b \\ a &= \frac{\delta_{lin}(N_{ts}) - \delta_{lin}(0)}{N_{ts}} \\ b &= \delta_{lin}(0) \end{aligned} \quad (9)$$

The third function is an exponentially decaying function, with a as the initial tolerance and r as the rate decay.

$$\begin{aligned} \delta_{expo}(t) &= a(1-r)^t \\ a &= \delta_{expo}(0) \\ r &= 1 - \left(\frac{\delta_{expo}(N_{ts})}{\delta_{expo}(0)}\right)^{\frac{1}{N_{ts}}} \end{aligned} \quad (10)$$

In the experiments, N_{ts} is set to $200,000$ with total number of timesteps for training set to $500,000$, the initial tolerance $\delta(0)$, to 20.0 mm and the final tolerance $\delta(N_{ts})$, to 1.0 mm. The initial tolerance was chosen based on evaluation errors during training of previous work [11]. 20.0 mm is approximately where improvement in the policy begins to decelerate. $200,000$ steps was chosen to have the remaining $300,000$ steps to train with the final goal tolerance. The number of total training steps has been significantly reduced from 2 million from previous work.

Tube	1^{st}	2^{nd}	3^{rd}
Precurvature (m^{-1})	21.3	13.1	0.1
Inner Diameter (mm)	0.7	1.4	2.0
Outer Diameter (mm)	1.1	1.8	2.2
Straight Length (mm)	431	332	10
Curved Length (mm)	103	113	5
Young's Modulus (GPa)	6.4	5.3	4.7
Shear Modulus (GPa)	2.5	2.2	3.0

TABLE I: Concentric tube robot tube parameters based on [13].

C. Proprioceptive and Egocentric Joints Representation

Additionally, the use of egocentric or relative joint representation rather than proprioceptive or absolute joint representation is investigated. In the egocentric representation the current joint variable is relative to the previous joint, with the first joint is referenced from the base reference frame as opposed to a proprioceptive representation where all joints are referenced from a base reference. For rotations,

$$\alpha_{relative} = \{\alpha_1, \Delta\alpha_{2-1}, \Delta\alpha_{3-2}\} \tag{11}$$

and extensions

$$\beta_{relative} = \{\beta_1, \Delta\beta_{2-1}, \Delta\beta_{3-2}\} \tag{12}$$

To retrieve the absolute joint representation, the cumulative sum is taken as shown below:

$$\alpha_{absolute} = \{\alpha_1, \Delta\alpha_{2-1} + \alpha_1, \Delta\alpha_{3-2} + \Delta\alpha_{2-1} + \alpha_1\} \tag{13}$$

The analysis of different representations is motivated by the use of egocentric joint representations in most continuous control tasks as shown in the DeepMind control suite [21]. Moreover, in an in-depth study at improving human-like motion with RL and imitation learning, egocentric over non-egocentric representation was chosen [22].

D. Simulation Environment

To simulate the kinematics of the CTR, as well as visualize the backbone shape and tubes of the robot, the geometrically exact modelling technique was chosen [8] in this work. In previous RL work [11] for CTRs, a simplified piecewise dominant stiffness model with constant curvature based on [7] was used to enable the exploration study. The simulation environment follows the framework of openAI gym [23] with the modelling technique used to compute end effector position or achieved goal, retrieve new desired goals with sampled joint configurations and visualise the robot. A three tube CTR system was chosen to train on as single and two tube systems have analytical solutions available and four tube systems have restricted use due to the limited workspace available from torsional windup snapping and instability [7]. Tube parameters found in Table I are based on a three tube hardware system [13].

E. Policy Learning

For the policy, a multilayer perceptron architecture of input size that of the state dimension and output the size of action dimension was used. With the MDP previously defined, any known DRL approach that is compatible with continuous state and actions and is off-policy can be incorporated to train the policy. Deep deterministic policy gradient (DDPG) [24] with hindsight experience replay (HER) [20] was chosen because DDPG has been shown to be more stable than other algorithms in stable environments [25]. HER with the future goal sampling strategy was added as in a environment where successful trajectories are sparse, the ability to add relabelled failed trajectories into successful ones is important for training convergence. Exploration noise was added to actions in the form of multivariate Gaussian noise as shown in [11].

IV. EXPERIMENTS AND RESULTS

For all training, a server cluster with Intel Gold 6130 18C 140W 2.3 GHz with 19 parallel [20] workers for 500,000 training steps with stable baselines [26] was used. We compare combinations of proprioceptive and egocentric representation with the three goal-based curriculum functions: constant, linear and exponential decay. Following this, path following experiments are performed with a noiseless and noise-induced environment with Gaussian noise added to the state. Specifically, noise is added to the joint rotation and extension of each tube as well as the computed end-effector or achieved goal position. We compare mean errors in the path in a noise-induced environment with a non-robust and robust policy. A robust policy is one where the noise-induced environment is used as the training environment.

First, to compare proprioceptive and egocentric representations, we plot the mean errors and success rate of the constant curriculum with both representations as shown in Fig. 4. We define success rate as the number of successful episodes over the total number of evaluation episodes with the learned policy at that training step. The egocentric representation (red) is able to perform better with a faster convergence and lower overall final error of 2.06 mm as compared to proprioceptive (blue) final error of 3.24 mm. From previous CTR RL work [11], multivariate Gaussian noise in action exploration in training is also shown for contrast in green. There is a large gap in performance due the starting configuration at the beginning of training. The robot is set to full extension, where in the previous work, the robot is set to full retraction and requiring a specific exploration strategy for extension actions.

Next, to compare the curriculum functions, we present errors and success rate of the constant, linear and decay curricula with the egocentric representation. The main advantage, as shown in Fig. 5, is faster convergence in the initial $200,000$ steps of training. As the goal tolerance starts large, desired goals are easily achieved by the agent. As a result, success rate of the linear and decay curriculum are high within the first $10,000$ steps. Linear and decay curricula also reduce errors quicker as compared to the constant

	Curriculum	Error (mm)	Var	Success rate
Proprio-ceptive	**Constant**	2.43	0.09	0.85
	Linear	3.31	0.15	0.87
	Decay	3.16	0.12	0.80
Ego-centric	**Constant**	1.97	0.05	0.87
	Linear	3.38	0.15	0.89
	Decay	1.29	0.03	0.93

TABLE II: Table of evaluation results for error, variance and success rate for representation and curriculum combinations.

curriculum. Aside from the training convergence benefits, higher success rates will reduce training times, as failed episodes will take longer to complete than successful ones. Illustrative examples of a successful and a failed trajectories of egocentric decay and proprioceptive constant functions at $125,000$ training steps are shown in Fig. 6. Although the egocentric decay policy in Fig. 6b seems to take unnecessary actions in this the early policy, it is able to achieve the desired goal despite the large distance. The proprioceptive policy in Fig. 6a on the other hand, where the policy has not had many successful episodes for training, is unable to reach a closer desired goal. To summarize the curriculum results, the final learned policy is evaluated on 1000 episodes with randomly sampled joint configurations and desired goals at each episode with the mean errors and success rate presented in Table II.

An additional possibility with RL and a HER based approach is the ability to transition to new desired goals. Since the desired goal is included in the state, as shown in (5), it can be changed after each timestep or episode. We set a trajectory generator to update the desired goal after the completion of a 20 timestep episode with no feedback of success or failure from the agent with respect to previously set goals. This inherent controller in RL, where the goal can be updated in the state itself, is unavailable to neural network approaches mentioned in §II. We use this open-loop controller to set straight line paths and circular paths as shown in Fig. 7 and Fig. 8. First, we employ an agent to follow a straight line in the simulation environment. In Fig. 7, the full path is shown through Fig. 7a to Fig. 7c.

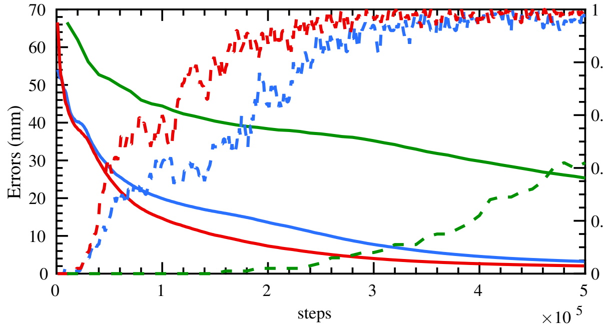


Fig. 4: Constant curriculum training results for joint proprioception (blue) and egocentric (red) and multivariate Gaussian exploratory noise (green) from previous RL work [11]. Errors are a solid line and success rate are dashed lines.

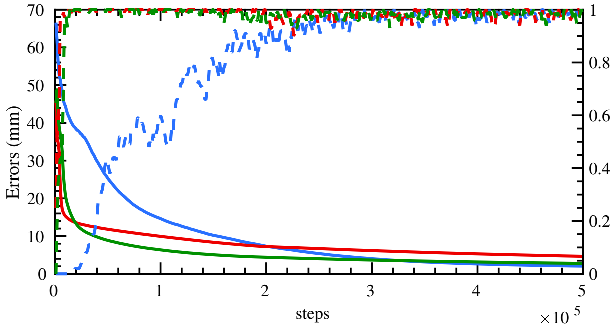
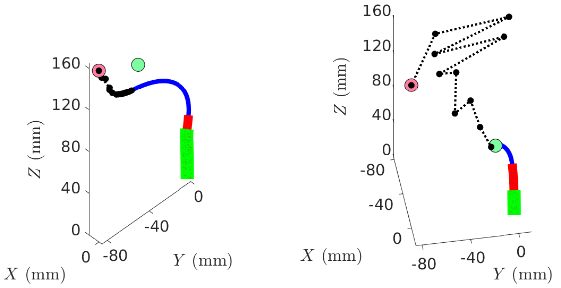


Fig. 5: Constant (blue) linear (red) and decay (green) curriculum with egocentric representation. Errors are a solid line and success rate are dashed lines.



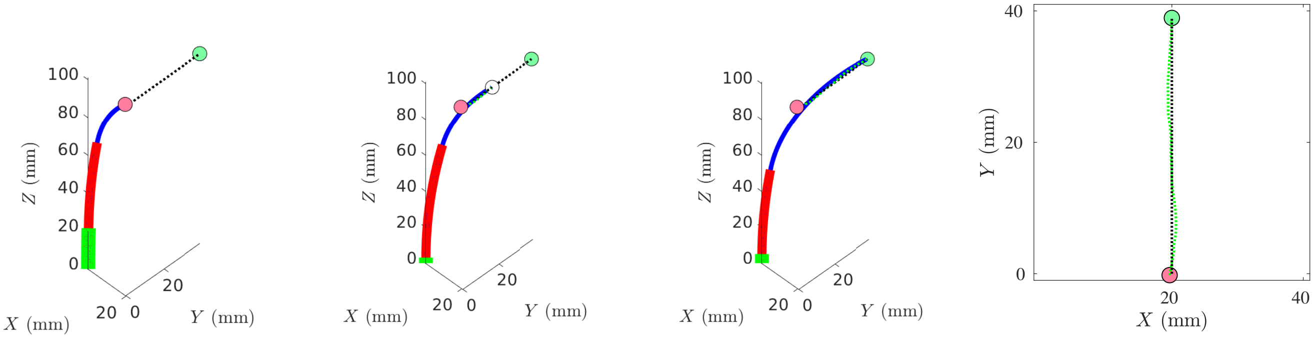
(a)

(b)

Fig. 6: Failed proprioceptive constant curriculum (left) and successful egocentric decay curriculum (right) IK solutions at 125k training steps. Red dot is starting position, green is desired goal, black dotted line is achieved trajectory.

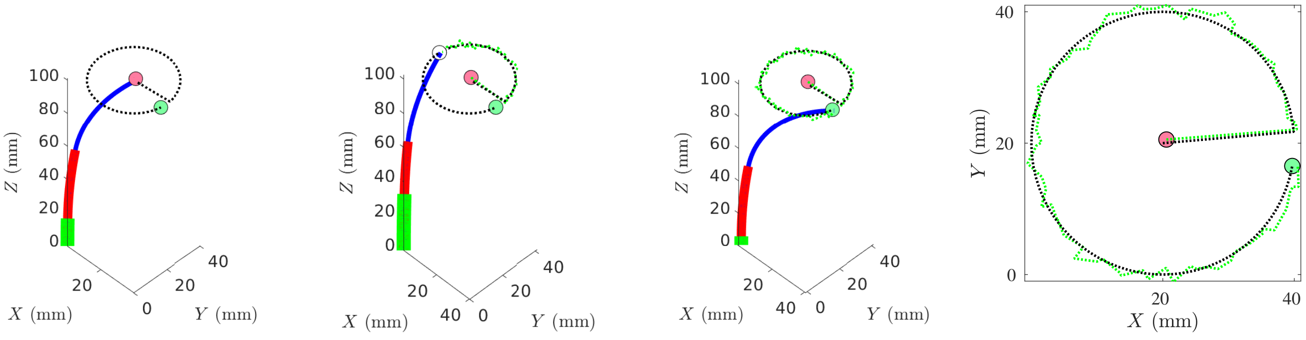
In Fig. 7d a top-down view supplements visualization of the path followed. The best performing policy of egocentric representation with a decay curriculum was used as the agent. The agent was able to follow the path accurately with a mean tracking error of 0.58 mm.

To begin experimentation in hardware, a robust policy that can deal with the inherent complexities and disturbances must be developed. To demonstrate this, a second simulation environment was created where noise was induced in the state. Specifically, zero mean Gaussian noise was added to the joint configuration (β and α) as encoder noise and the achieved goal position as tracking noise. Encoder noise is variable and depends on a large number of factors. For simplicity, a 1° standard deviation was selected. To determine the extension joint noise, a gear ratio of 0.001 was used. For achieved goal or tracking noise, a standard deviation 0.8 mm is used based on an EM tracker (Aurora, NDI Inc., CA) precision in documentation. With the noise-induced environment, we train a new robust egocentric decay curriculum policy. To compare the new robust policy to a non-robust policy we compute the mean error in a circular path following task as shown in Fig. 8. The robust policy and non-robust policy had a mean tracking error of 1.37 mm and 1.56 mm respectively. Although this is only a marginal



(a) Start of trajectory.
(b) Middle of trajectory.
(c) Trajectory complete.
(d) X-Y view of full trajectory.

Fig. 7: Trained egocentric decay policy on simulation environment following straight trajectory. Red dot is starting position, green is the end position, black dashed line is desired trajectory and green is achieved trajectory.



(a) Start of trajectory.
(b) Middle of trajectory.
(c) Trajectory complete.
(d) X-Y view of full trajectory.

Fig. 8: Trained egocentric decay robust policy on noise-induced simulation environment following circular trajectory with achieved trajectory (green dashed line) and desired trajectory (black dashed line).

improvement, we aim to utilise the various sim2real transfer learning approaches available literature in the future to build on this result. In the attached video, trajectory animations for example IK solutions for proprioceptive and egocentric agents at 125k training steps are provided. Also, example IK solutions for the egocentric decay curriculum are shown. In the last section of the video, animations for path following of circle and line trajectories with an without a robust policy is shown.

Our main comparison is to the previous RL work [11], where an error 0.5 mm Cartesian error for IK solution was reported with a simplified dominant stiffness constant curvature model and shorter overall tubes lengths. In this work, using a more accurate simulation environment we report 1.29 mm mean Cartesian error in IK evaluation using egocentric decay curriculum and 1.37 mm mean tracking error with a robust policy in a noise-induced simulation. In a noise-free path following task, we achieve 0.58 mm error. In terms of training, overall training steps have been reduced from 2 million to $500,000$. In the recent MPC work by Khadem et al. [13], they report 0.203, 0.123 and 0.080 mm tracking error for a Jacobian-based approach, and the MPC approach with $K = 2$ and $K = 5$ in simulation. Although the tracking errors are higher, this preliminary work can be improved in numerous ways. Determining optimised goal tolerance parameters (N_{ts}, $\delta(0)$, $\delta(N_{ts})$) as well as increasing number of training steps should improve error metrics. Also, investigating error propagation through rotation and goal distances may reveal further improvements by identifying areas of under-exploration.

V. CONCLUSIONS

In this paper, DRL with a goal-based curriculum was used to improve training convergence and robustness of learned policies to control CTRs. Specifically, linear and exponentially decaying goal tolerances combined with an egocentric joint representation showed significant improvement in on a more accurate model when compared to previous work that used a simple constant curvature model. Moreover, the curriculum presented was used to train a robust policy within a noise-induced simulation environment.

In future work, we plan to incorporate the backbone shape in the state representation and fully validating our work in hardware. We also aim to investigate errors in rotation as the accuracy of this method depends on full workspace exploration and error propagation through goal distances. Incorporating regions that are unexplored could greatly improve accuracy results. Lastly, varying curricula parameters, especially the final goal tolerance $\delta(N_{ts})$, could lead to further improvement in training and error results.

REFERENCES

[1] Hunter B Gilbert, D Caleb Rucker, and Robert J Webster III. Concentric tube robots: The state of the art and future directions. *Robotics Research*, pages 253–269, 2016.

[2] F. Lin, C. Bergeles, and G. Yang. Biometry-based concentric tubes robot for vitreoretinal surgery. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5280–5284, 2015.

[3] R. Wirz, L. Torres, P. Swaney, H. Gilbert, R. Webster R. Alterovitz, K. Weaver, and P. Russell. An experimental feasibility study on robotic endonasal telesurgery. *Neurosurgery*, 76 4:479–84; discussion 484, 2015.

[4] G. Dwyer, F. Chadebecq, C. Bergeles M. Amo, E. Maneas, V. Pawar, E. Vander Poorten, J. Deprest, S. Ourselin, and P. De Coppi. A continuum robot and control interface for surgical assist in fetoscopic interventions. *IEEE robotics and automation letters*, 2(3):1656–1663, 2017.

[5] G. Dwyer, R. J. Colchester, E. J. Alles, E. Maneas, S. Ourselin, T. Vercauteren, J. Deprest, E. V. Poorten, P. D. Coppi, A. E. Desjardins, and D. Stoyanov. Robotic control of a multi-modal rigid endoscope combining optical imaging with all-optical ultrasound. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3882–3888, 2019.

[6] J. Burgner-Kahrs, D. C. Rucker, and H. Choset. Continuum robots for medical applications: A survey. *IEEE Transactions on Robotics*, 31(6):1261–1280, 2015.

[7] P. Dupont, J. Lock, B. Itkowitz, and E. Butler. Design and control of concentric-tube robots. *IEEE Transactions on Robotics*, 26(2):209–225, apr 2010.

[8] D. C. Rucker, B. A. Jones, and R. J. Webster III. A geometrically exact model for externally loaded concentric-tube continuum robots. *IEEE Transactions on Robotics*, 26(5):769–780, 2010.

[9] Jesse Lock and Pierre E Dupont. Friction modeling in concentric tube robots. In *2011 IEEE International Conference on Robotics and Automation*, pages 1139–1146. IEEE, 2011.

[10] Michele Giorelli, Federico Renda, Marcello Calisti, Andrea Arienti, Gabriele Ferri, and Cecilia Laschi. Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature. *IEEE Transactions on Robotics*, 31(4):823–834, 2015.

[11] K.Iyengar, G.Dwyer, and D.Stoyanov. Investigating exploration for deep reinforcement learning of concentric tube robot control. *International Journal of Computer Assisted Radiology and Surgery*, 15(7):1157–1165, June 2020.

[12] J. Burgner, D. C. Rucker, H. B. Gilbert, P. J. Swaney, P. T. Russell, K. D. Weaver, and R. J. Webster. A telerobotic system for transnasal surgery. *IEEE/ASME Transactions on Mechatronics*, 19(3):996–1006, 2014.

[13] M. Khadem, J. O'Neill, Z. Mitros, L. da Cruz, and C. Bergeles. Autonomous steering of concentric tube robots via nonlinear model predictive control. *IEEE Transactions on Robotics*, 36(5):1595–1602, 2020.

[14] Mohamed Nassim Boushaki, Chao Liu, and Philippe Poignet. Task-space position control of concentric-tube robot with inaccurate kinematics using approximate jacobian. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5877–5882. IEEE, 2014.

[15] C Bergeles, FY Lin, and GZ Yang. Concentric tube robot kinematics using neural networks. In *Hamlyn symposium on medical robotics*, pages 1–2, 2015.

[16] R. Grassmann, V. Modes, and J. Burgner-Kahrs. Learning the Forward and Inverse Kinematics of a 6-DOF Concentric Tube Continuum Robot in SE(3). In *IEEE International Conference on Intelligent Robots and Systems*, pages 5125–5132. Institute of Electrical and Electronics Engineers Inc., dec 2018.

[17] Reinhard Grassmann and Jessica Burgner-Kahrs. On the merits of joint space and orientation representations in learning the forward kinematics in se (3). In *Robotics: science and systems*, 2019.

[18] A. Kuntz, A. Sethi, R.J Webster, and R. Alterovitz. Learning the Complete Shape of Concentric Tube Robots. *IEEE Transactions on Medical Robotics and Bionics*, 3202(c):1–1, 2020.

[19] R. Sutton. and A. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[20] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. In *Advances in neural information processing systems*, pages 5048–5058, 2017.

[21] Y. Tassa, S. Tunyasuvunakooland A. Muldal, Y. Doron, S. Liu, S. Bohez, T. Erez J. Merel, T. Lillicrap, and N. Heess. dm_control: Software and tasks for continuous control, 2020.

[22] J. Merel, Y. Tassa, TB Dhruva, S. Srinivasan, Jay Lemmon, Ziyu Wang, G. Wayne, and N. Heess. Learning human behaviors from motion capture by adversarial imitation. *ArXiv*, abs/1707.02201, 2017.

[23] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016.

[24] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, NT. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[25] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters. *arXiv preprint arXiv:1709.06560*, 2017.

[26] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O.Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu. Stable baselines. github.com/hill-a/stable-baselines, 2018.