A Learning-based Approach for Adaptive Closed-loop Control of a Soft Robotic Arm

Francesco Piqué, Hari Teja Kalidindi, Arianna Menciassi, Cecilia Laschi, Egidio Falotico

***Abstract*—The characteristic compliance of soft/continuum robot manipulators entails them with the desirable features of intrinsic safety, low power to actuation ratio and adaptability to the environment. At the same time, it makes analytical models excessively slow for efficient use in control. We propose a recurrent neural network (RNN) approach for adaptive model based closed-loop control of a continuum robot. First, the forward dynamic model is trained offline on data obtained by continuous motor babbling, learning the relationship between the actuators' inputs and the robot tip position. Then another network, named inverse model, is used as a closed loop controller and trained by minimizing the forward model tracking error. We show that using the trained controller, the continuum robot is able to track a circular task with a low RMS error, and to maintain its performance under an external load, after updating the networks' weights.**

***Index Terms*—soft robotics, control, learning**

I. Introduction

Unlike industrial robots, which are equipped with rigid links and can be modeled with closed form equations and with a finite number of parameters, soft and continuum robots are hard to model due to under-actuation, redundancy and hysteresis of materials [1]. The most well known modeling approximation for soft robot control is the constant curvature (CC) assumption, which by assuming a constant curvature an all the length of the soft manipulator and by ignoring the dynamics of the robot, enables steady state control with low computational cost [2]. Joining several constant curvature sections provides the piecewise constant curvature assumption, which also enables soft robotic control with relative ease [3]. However these model based methods tend to fail when the soft robot is highly nonlinear, non-uniform and subject to external uncertainties [1]. Learning based approaches are an optimal solution since they do not require a priori knowledge of the manipulator dynamics, which can be derived by supervised learning. The first attempt to use a feed-forward neural network component for soft robot control was published in [4]. Several more recent works have shown the capability of neural networks for both open-loop and closed-loop control of soft robots [5] [6]. In this work we propose an adaptive control framework based on recurrent neural networks (RNN). Our contribution is a control framework which is adaptive, since the neural networks of which it is composed can update their parameters to accommodate for external unmodeled perturbations. We are able to control a circular trajectory tracking experiment on a real soft robotic manipulator. We apply an external load to the manipulator, and show that, after a retraining phase, the control is able to adapt itself to the perturbation introduced by the load, and to maintain the previous tracking performance.

All authors are with The Biorobotics Insitute, Sant'Anna School of Advanced Studies; e-mail: {f.pique, h.kalidindi, a.menciassi, c.laschi, e.falotico@santannapisa.it}

II. Methods

A soft robotic manipulator for the assistance of elderly people has been used in this work [7]. It is composed of three soft modules, each actuated by means of three pneumatic McKibben actuators. The McKibben actuators are controlled by pneumatic valves (Camozzi K8P) which are controlled by a simple PID implemented on an Arduino Due board. The Arduino is in turn controlled by a PC using a serial port. In this work, the proximal module of the robot has been selected and used independently of the others.

Input/output data were acquired by means of continuous motor babbling. The inputs are chosen randomly in the 0 to 100 value range (corresponding 0 to 0.83 bar) so as not to risk damage to the actuator, keeping every actuator value in a ± 40 bit range of the previous value. This range is chosen as a trade-off between avoiding motor saturation (the pneumatic chamber is able to reach the desired pressure before the next actuation sample) and having a sufficiently explored workspace (a low actuator range means more samples are needed for exhaustive exploration) [5]. Since the relationship between the applied pressures and the commanded values is quite linear, the latter are considered directly as input data. Conversely, the position of the tip of the manipulator, measured with an electromagnetic tracking system (NDI Aurora ®), is considered as output data. A dataset of 10000 samples was acquired at a frequency of 10 Hz and split in training set (7000 samples) and test set (3000 samples).

The forward model of the manipulator is built as the mapping between the current control input u^t, containing the commanded values for the pneumatic actuators, the two previous end-effector positions x^{t-1} and x^{t-2}, and the current end effector position x^t (Fig. 1). The network consists of a RNN layer of 100 neurons followed by a linear layer and a $tanh$ layer. The model's weights are optimized on the training set by supervised learning for 1200 epochs, with a mean square error (MSE) loss function and a learning rate of 10^{-4}. After 1200 training epochs the RMS error is as low as 3 mm.

The forward model is then exploited to learn an inverse model, which acts as a feedback controller of the forward model (Fig. 1). The controller is a neural network of the same structure as the forward model, with a sigmoid layer instead of a $tanh$ layer, to prevent actuator saturation. The controller is

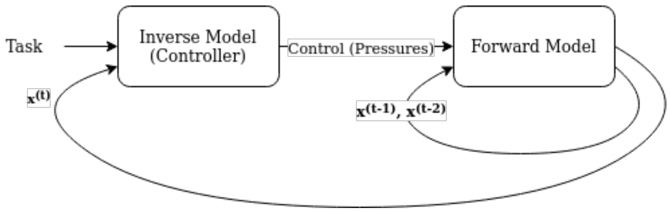

Fig. 1: The proposed adaptive control architecture.

TABLE I: RMS Error

Experiment	X [mm]	Y [mm]	Z [mm]
Simple tracking	6	17	2
External load	23	17	11
External load with retraining	14	13	5

trained to have the forward model follow a circular task. The loss function is the MSE between the forward model output x^t and the desired task. Finally the real robot is substituted to the forward model, for testing the trained controller by carrying out a circular trajectory tracking experiment.

To assess the adaptability of the control, an external load of 50 g was applied to the manipulator. Firstly we use the previously trained controller to track a circular trajectory using the manipulator with the applied load, while simultaneously recording the robot's tip position. The obtained task space error is used to update the weights of the previously trained forward model. The weights are updated by supervised learning for 100 epochs using only the data acquired by performing the tracking task with the external load, minimizing the MSE error between the target circle and the real robot tip position. A weight decay of 10^{-5} is introduced to prevent over-fitting, measured with the electromagnetic tracking system. Similarly, the inverse model (controller) is updated on the newly trained forward model. The newly trained controller is finally used to have the robot with the added weight follow the same circular task.

III. Results

We show that, by using the proposed control architecture, the soft robot is able to track a circular trajectory with an RMS error as low as 6 mm on the x-axis, 17 mm on the y-axis and 2 mm on the z-axis. Once the external load is applied, it causes the robot to deviate from the previously learnt trajectory, with an increase of the RMS error both on the x-axis and on the z-axis. After updating the weights of the forward and of the inverse model, the tracking performance of the manipulator improves an all axes, tracing a trajectory similar to the one of the tracking without weight experiment, making the proposed controller adaptive (Fig. 2). Although the accuracy in the simple tracking experiment can be improved, it is significant to note the ability of the control to improve its accuracy by retraining, after applying an external unmodeled load. Table I shows the RMS error of the experiments described above.

.

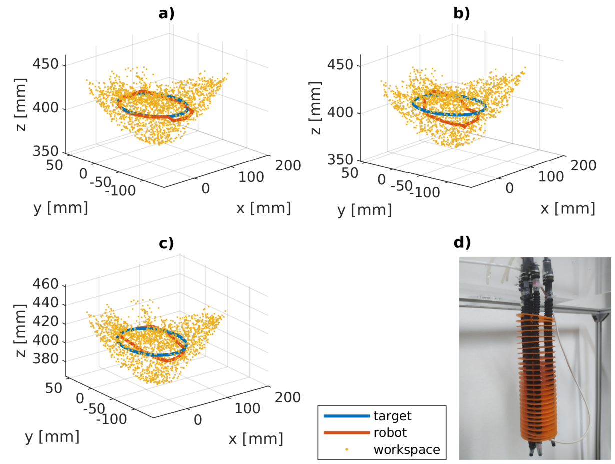

Fig. 2: Circular tracking experiment a) without external load, b) with external load, c) with external load after model retraining. In d) the soft robot used for the experiments is shown.

IV. Conclusion

We have proposed a control framework based on recurrent neural networks for soft/continuum robots. This framework enables trajectory tracking for soft manipulators, and is suitable for tracking under disturbances such as external loads, after a retraining phase. The system is applicable also to different soft manipulators and can be used for future work in learning based approaches for control of soft robots.

V. Acknowledgement

The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Programme in the framework of PROBOSCIS Project (GA No. 863212).

References

[1] T. G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: a survey," *Soft robotics*, vol. 5, no. 2, pp. 149–163, 2018.

[2] B. A. Jones and I. D. Walker, "Kinematics for multisection continuum robots," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 43–55, 2006.

[3] C. Della Santina, R. K. Katzschmann, A. Biechi, and D. Rus, "Dynamic control of soft robots interacting with the environment," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 46–53.

[4] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, "A neural network controller for continuum robots," *IEEE transactions on robotics*, vol. 23, no. 6, pp. 1270–1277, 2007.

[5] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Learning dynamic models for open loop predictive control of soft robotic manipulators," *Bioinspiration & biomimetics*, vol. 12, no. 6, p. 066003, 2017.

[6] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 124–134, 2018.

[7] M. Manti, A. Pratesi, E. Falotico, M. Cianchetti, and C. Laschi, "Soft assistive robot for personal care of elderly people," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pp. 833–838, IEEE, 2016.