

Intelligent Gain Adaptation for Model-Free Control of Tendon-Driven Continuum Robotic Arms using Fuzzy Inference System

Nima Maghooli, Omid Mahdizadeh and S. Ali A. Moosavian

Center of Excellence in Robotics and Control, Advanced Robotic and Automated Systems (ARAS)

Department of Mechanical Engineering, K. N. Toosi University of Technology, Tehran, Iran;

Emails: nima.maghooli@ut.ac.ir, omid.mahdizadeh@email.kntu.ac.ir, moosavian@kntu.ac.ir.

Abstract – Continuum manipulator modeling is always associated with structured and unstructured uncertainties, and model-based control system design for this class of robotic systems will be very challenging. On the other hand, the performance quality of model-free controllers is completely dependent on their hyperparameters, and they are very sensitive to the trajectory considered for the system. As a result, using these methods will require setting parameters for different scenarios. The main goal of this research is to design a model-free controller with similar performance to model-based control strategies. At the same time, hyperparameters are determined online by the gain adjustment system. The Transpose Jacobin (TJ) algorithm is one of the simplest and most widely used model-free control algorithms in the control of robotic systems. TJ modified version (MTJ) by estimating feedback linearization performs very well in trajectory tracking, and due to the use of the PID controller structure, it is considered a local robust control strategy. Since TJ does not have an error equation and MTJ also has an approximate error equation, an adaptive gain adjustment system can greatly increase the algorithms' potential and establish the ability to follow trajectories in different work points in the system workspace. For this reason, this research tries to improve the performance of TJ and MTJ model-free control strategies in following different trajectories, starting from arbitrary initial conditions in the system's workspace. The research innovation is to use a supervised machine learning method (fuzzy inference system) to implement the intelligent gain adaptation system to achieve this goal.

Keywords – *Continuum Manipulator, Modified Transpose Jacobian (MTJ), Fuzzy Inference System (FIS), Intelligent Gain Adaptation*

I. INTRODUCTION

Continuum robotic systems are a group of robots made of flexible materials, generally have narrow shapes, and have a very high ability to perform complex three-dimensional movements. Miniaturization capacity, high skill, and the absence of rigid links and joints have led to the attention of this category of robots for various applications, including medical applications. Their size characteristics, the differences in modeling, and the complexity in route design led to the emergence of multiple approaches to control these systems [1]. According to the definition, the continuum robot is considered an actuatable structure whose constituent materials provide the ability to create curves with continuous tangent vectors. They have unique features such as elastic structure, infinite degree of freedom (theoretically), and continuous bending capability. Continuum robots are considered a subset of hyper-redundant robotic systems, for which the definition of continuum can also be considered true for these systems. But in fact, they have a large number of

separate and rigid links and joints, and continuous robots have been developed to upgrade and evolve these systems and increase their use. They have been used in a variety of applications, including medical (minimally invasive surgery), manufacturing, aerospace, search and rescue, and nuclear [2]. In different references, different classifications with different points of view are presented for robotic systems. In Fig. 1, robots are classified in terms of structure (materials used in their mechanical structure) and degrees of freedom, and according to the classification, the position of continuum robots in this categorization is specified [3]. This research focuses mainly on the Tendon-Driven Continuum Robots (TDCRs).

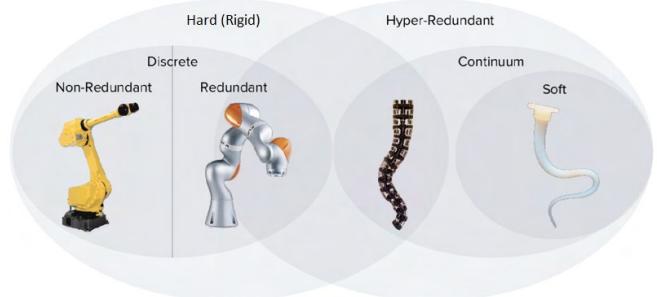


Fig. 1. Classification of robots in terms of structure and degrees of freedom [3]

The Transpose Jacobian (TJ) controller is a common model-free control algorithm that performs well in controlling robotic systems' positions despite its simple structure. It has advantages such as simplicity and being model-free, but also disadvantages that motivate its modification. It is sensitive to noise due to its large control gains, and its tracking performance decreases for fast trajectories, and there is no systematic method to determine its control gains [4]. Fig. 2 shows a customized TJ block diagram.

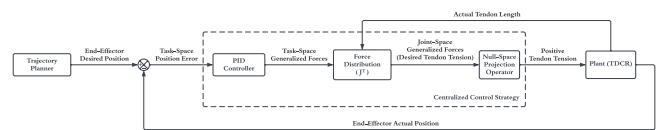


Fig. 2. Customized TJ block diagram for TDCRs

To address the disadvantages of the TJ controller, the MTJ controller is proposed, which aims to modify the TJ control strategy while maintaining its advantages. The MTJ controller implements a function similar to feedback linearization in model-based controllers by estimating the

system dynamics using the previous control input [5]. The MTJ controller is asymptotically stable and is proven using Lyapunov's stability theorems [6].

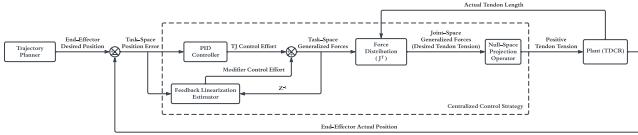


Fig. 3. Customized MTJ block diagram for TDCRs

Fuzzy Inference System (FIS) has significant potential for solving engineering problems, particularly in the control of continuum manipulators [7–9]. As another bonus for the fuzzy inference system, we can mention the possibility of combining it with other heuristic methods like reinforcement learning [10]. Almost most of the work done in this field is kinematic control [11, 12]. The main goal of this research is to provide closed-loop dynamic control for tendon-driven continuum robotic arms. Generally, three strategies can be considered for applying FIS in the control loop [13]. The first strategy uses the FIS as an intelligent controller and force distributor. The block diagram for this strategy is shown in Fig. 3. This strategy does not require an explicit model of the system's dynamics and kinematics.

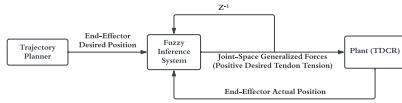


Fig. 4. Proposed block diagram for the 1st Strategy

The second strategy uses the FIS as an intelligent controller. This strategy does not require an explicit model of the system's dynamics but uses the system's kinematics (Jacobian matrix) to establish force distribution. The block diagram for this strategy is shown in Fig. 4.

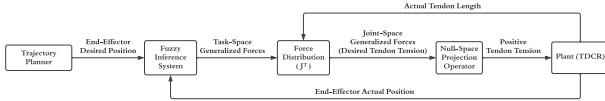


Fig. 5. Proposed block diagram for the 2nd Strategy

The third strategy uses the FIS as an adaptive gain tuner. It also does not require an explicit model of the system's dynamics but uses its kinematics (Jacobian matrix) for force distribution. The block diagram for this strategy is shown in Fig. 5.

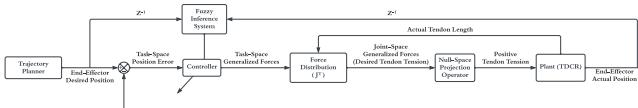


Fig. 6. Proposed block diagram for the 3rd Strategy

Two critical considerations lead to the selection of the third strategy. Firstly, deriving the forward kinematics equations for obtaining the Jacobian matrix is relatively easy, making the first strategy less appealing. Secondly, using the FIS as an intelligent controller requires a longer time to design membership functions and rule bases. This is less practically viable due to the increased design time and computational cost. The third strategy offers several advantages, including:

- Reduced design time since the FIS does not need to acquire the controller's structure.
- The ability to develop robust controllers with adaptive gains of the controller.
- A lightweight and suitable gain adaptation system for real-time implementation on hardware.
- Ensured stability in the design process. An information range for controller gains minimizes issues related to instability and other challenges during the design process.

Ultimately, the third strategy aligns with the primary objective of designing a controller with variable and suitable gains for tracking any desired path within the system's workspace [14].

This paper is organized as follows: a brief explanation of dynamics modeling used in simulations and control system design is presented in section 2. Following that, section 3 proposes a control system design using a fuzzy inference system. To verify the results, the calibration process for a real platform is discussed in section 4. The results of software simulations in both ideal and realistic (in the presence of noise and disturbance) conditions and implementations on a physical platform are presented in section 5. The last section of this paper summarizes the conclusions achieved in this research.

II. DYNAMICS MODELING

In this research, a static model with the assumption of constant curvature for each sub-segment is used, commonly referred to as the "piecewise constant curvature model" [15]. This model is computationally less expensive than a variable curvature model while maintaining the necessary accuracy. The variable curvature model considers a time-, shape-, and tendon tension force-dependent function ($r = f(t, s, T)$) and the momentary position of each point in the backbone will offer with exceptionally high precision. However, this assumption leads to a set of nonlinear partial differential equations, which are computationally intensive and may not be practically feasible. On the other hand, the piecewise constant curvature model simplifies the problem by introducing two simplifying assumptions, effectively eliminating the need for both time and spatial dependency in the model. This simplification results in a set of nonlinear algebraic equations, which significantly reduces the computational load. Moreover, the accuracy of this model remains satisfactory when compared to the variable curvature model [16]. Like other robotic systems, the kinematics and dynamics equations (forward and inverse) expressed in the position control problem are shown in Fig. 6.

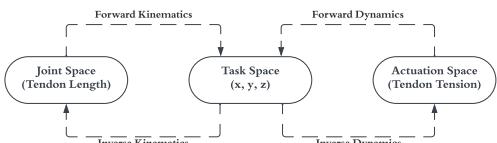


Fig. 7. Kinematics and dynamics of TDCRs in position control problem

III. CONTROL SYSTEM DESIGN

Assuming that the end-effector's position vector is $\mathbf{X} = [x \ y \ z]^T$ and the effective length vector of the tendons, denoted by $\mathbf{L} = [l_1 \ l_2 \ l_3 \ l_4 \ l_5 \ l_6]^T$, then the linear Jacobian matrix ($\mathbf{J} = \mathbf{J}_L \dot{\mathbf{L}}$) that relates the rate of change of

these two vectors is calculated as follows, according to the explanation given in [17].

$$J_{Lnm} = \frac{\partial x_n}{\partial l_m} , n = 1, \dots, 3 , m = 1, \dots, 6 \quad (1)$$

Using equation (1), the Jacobian matrix is obtained as follows:

$$J_L = \begin{bmatrix} \frac{\partial x}{\partial l_1} & \frac{\partial x}{\partial l_2} & \frac{\partial x}{\partial l_3} & \frac{\partial x}{\partial l_4} & \frac{\partial x}{\partial l_5} & \frac{\partial x}{\partial l_6} \\ \frac{\partial y}{\partial l_1} & \frac{\partial y}{\partial l_2} & \frac{\partial y}{\partial l_3} & \frac{\partial y}{\partial l_4} & \frac{\partial y}{\partial l_5} & \frac{\partial y}{\partial l_6} \\ \frac{\partial z}{\partial l_1} & \frac{\partial z}{\partial l_2} & \frac{\partial z}{\partial l_3} & \frac{\partial z}{\partial l_4} & \frac{\partial z}{\partial l_5} & \frac{\partial z}{\partial l_6} \end{bmatrix} \quad (2)$$

The relationship between the generalized forces in the workspace $\mathcal{F} = [F_x \ F_y \ F_z]^T$ and the generalized forces in the joint space $\tau = [T_1 \ T_2 \ T_3 \ T_4 \ T_5 \ T_6]^T$, for control in the task space is expressed as following relation:

$$\tau = J_L^T \mathcal{F} \quad (3)$$

The control input vector is expressed in terms of the position error vector in the workspace, $e = [e_x \ e_y \ e_z]^T$, by the following relation:

$$\tau = J_L^T [K_p e + K_i \int e dt + K_d \dot{e}] \quad (4)$$

The gains in equation (4) are diagonal matrices and are given in the following form:

$$K_i = \begin{bmatrix} k_{i,x} & 0 & 0 \\ 0 & k_{i,y} & 0 \\ 0 & 0 & k_{i,z} \end{bmatrix} , i = P, I, D \quad (5)$$

The null space projection operator has been used to prevent tendons from loosening during movement, as described in [18]. The block diagram of the control system for the TJ strategy is shown in Fig. 9.

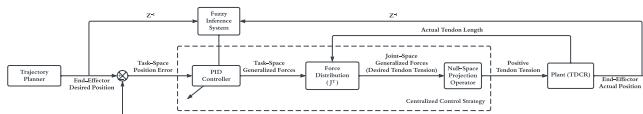


Fig. 8. Proposed Fuzzy-TJ block diagram for TDCRs

The TJ structure is modified by adding a modifier term like $\mathbf{h} = [h_x \ h_y \ h_z]^T$, to equation (4), therefore:

$$\tau = J_L^T [K_p e + K_i \int e dt + K_d \dot{e} + \mathbf{h}] \quad (6)$$

The modifier term, \mathbf{h} , is calculated as follows:

$$\mathbf{h}_{(t)} = K \mathcal{F}_{(t-\Delta t)} \quad (7)$$

In the MTJ strategy, the control input of the previous time step in the workspace is denoted by $\mathcal{F}_{(t-\Delta t)}$. The diagonal matrix K is defined as the following matrix:

$$K = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix} \quad (8)$$

The principal diagonal elements of K are calculated using the following relation:

$$k_i = \exp \left[- \left(\frac{|e_i|}{e_{max_i}} + \frac{|\dot{e}_i|}{\dot{e}_{max_i}} \right) \right] , i = x, y, z \quad (9)$$

In equation (9), e_{max_i} and \dot{e}_{max_i} are the error sensitivity threshold and the error derivative sensitivity threshold for activating the modifier term, respectively. The block diagram of the control system for the MTJ strategy is shown in Fig. 9.

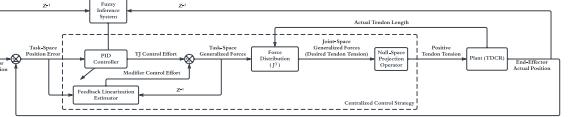


Fig. 9. Proposed Fuzzy-MTJ block diagram for TDCRs

As stated in the introduction, the fuzzy inference system has a high potential for application in solving engineering problems, especially control engineering. Fig. 10 shows an overview of the operation of the fuzzy inference system for application in control engineering problems.

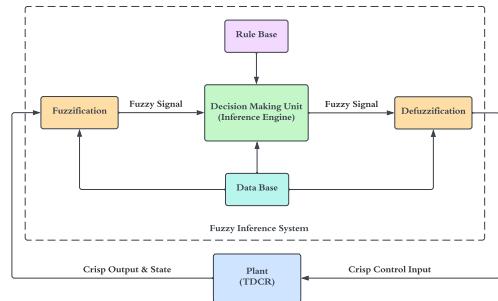


Fig. 10. Fuzzy Inference System applied to the control problem

The design stages of online tuning PID-based controllers using the fuzzy inference system, according to the explanations given in [19, 20], are as follows.

- **Definition of appropriate membership functions for fuzzy inference system inputs**

The membership functions considered for the inputs (error and its derivative) in the fuzzy inference system are of the Gaussian membership function type, shown in Fig. 11.

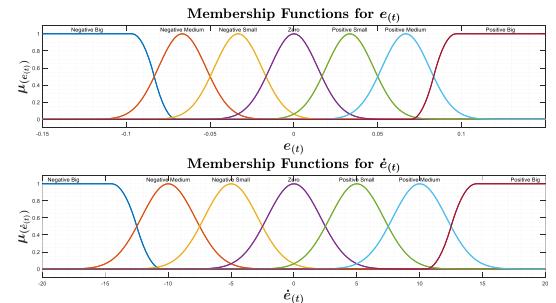


Fig. 11. Designated membership functions for FIS inputs

According to Fig. 11, the accumulation of membership functions in the range that has more importance (close to

zero) is greater. At distances further away from it, the last membership functions (large positive and large negative) continue with the maximum degree of membership, with the aim of covering all possible areas for the error and its derivative.

- Determining the appropriate range for K_P and K_D

The fuzzy inference system works based on the knowledge of the performance of the system, and here, to continue the design process, it is necessary to define a range for the K_P and K_D , and these gains must be in a certain range.

$$\begin{aligned} K_P &\in [(K_P)_{min}, (K_P)_{max}] \\ K_D &\in [(K_D)_{min}, (K_D)_{max}] \end{aligned} \quad (10)$$

Determining intervals like equation (10) is done by trial and error and testing different values for these gains. The gains K_I do not need to determine the interval, since it is not considered independent and its value is obtained according to the other gains.

- Defining suitable membership functions for fuzzy inference system outputs

The membership functions considered for the outputs (parameters needed to calculate the gains) in the fuzzy inference system are also considered Gaussian, as shown in Fig. 12.

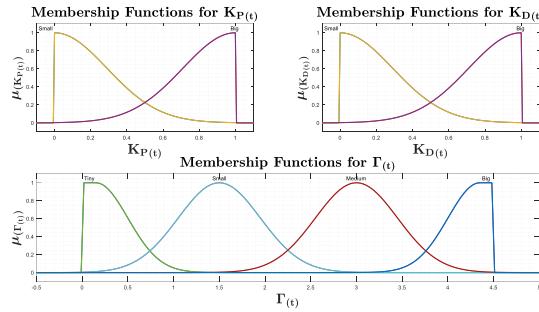


Fig. 12. Designated membership functions for FIS outputs

According to Fig. 12, only two membership functions have been used for the K_P and K_D gains, and four membership functions have been introduced for the Γ parameter. Here, unlike membership functions related to inputs, the target interval for each output is completely closed before the first and after the last membership function (that is, the degree of all membership functions outside the interval is zero). The output of the fuzzy inference system for K_P and K_D , numbers in the range $[0,1]$, and for Γ , the range $[0,4.5]$ is considered. Finally, to convert the normalized values into non-normalized numbers, according to the intervals considered for the gains, the following relationship is established:

$$\begin{aligned} K_P &= (K_P)_{min} + \bar{K}_P[(K_P)_{max} - (K_P)_{min}] \\ K_D &= (K_D)_{min} + \bar{K}_D[(K_D)_{max} - (K_D)_{min}] \end{aligned} \quad (11)$$

By finding the values of K_P , K_D and Γ , K_I is calculated by the following equation:

$$K_I = \frac{(K_P)^2}{\Gamma K_D} \quad (12)$$

- Definition of rule base of the fuzzy inference system

In Tables I to III, the rule bases in the fuzzy inference system for K_P , K_D and Γ are presented. Each input has 7 membership functions. As a result, the maximum number of rules is 49, as shown in the tables. The idea of writing these rules is taken from [19].

TABLE I. RULE-BASE FOR K_P

Rule-Base for $K_P(t)$	$e(t)$						
	NB	NM	NS	ZO	PS	PM	PB
$e(t)$	NB	B	B	B	B	B	B
	NM	S	B	B	B	B	S
	NS	S	S	B	B	S	S
	ZO	S	S	S	B	S	S
	PS	S	S	B	B	B	S
	PM	S	B	B	B	B	S
	PB	B	B	B	B	B	B

TABLE II. RULE-BASE FOR K_D

Rule-Base for $K_D(t)$	$\dot{e}(t)$						
	NB	NM	NS	ZO	PS	PM	PB
$e(t)$	NB	S	S	S	S	S	S
	NM	B	B	S	S	S	B
	NS	B	B	B	S	B	B
	ZO	B	B	B	B	B	B
	PS	B	B	B	S	B	B
	PM	B	B	S	S	S	B
	PB	S	S	S	S	S	S

TABLE III. RULE-BASE FOR Γ

Rule-Base for $\Gamma(t)$	$\dot{e}(t)$						
	NB	NM	NS	ZO	PS	PM	PB
$e(t)$	NB	T	T	T	T	T	T
	NM	S	S	T	T	T	S
	NS	M	S	S	T	S	S
	ZO	B	M	S	S	S	M
	PS	M	S	S	T	S	S
	PM	S	S	T	T	T	S
	PB	T	T	T	T	T	T

Finally, the obtained control surfaces, which show how the inputs and outputs of the fuzzy inference system are mapped, are shown in Fig. 13.

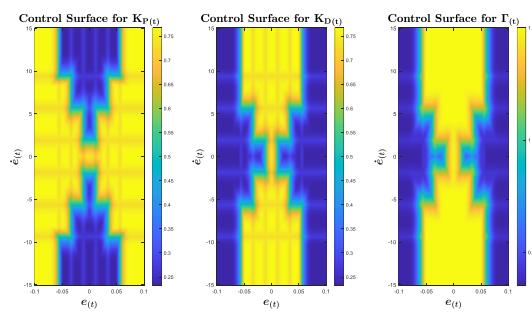


Fig. 13. Obtained control surfaces of FIS

IV. PHYSICAL SYSTEM CALIBRATION

In this section, the calibrations of the main sensors of the system, i.e., loadcell and camera, are explained.

A. Loadcells Calibration:

To calibrate the load cells, a tendon is connected to the loadcell, and its free end is guided outside the robotic arm chamber, keeping the same direction for the extension of the connecting tendon between the load cell and the connection point to the continuum robotic arm. Then, the tendon is connected to a hook by coupling, and 400 g weights (5 weights) are placed on the hook for each loadcell. After that, the digital number of the output of the load cell is read online using a laptop and recorded in the table.

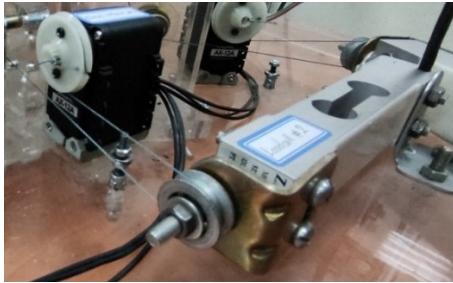


Fig. 14. Loadcell coupled with servo-motor

Finally, with digital numbers and their corresponding weight values, the most appropriate line is obtained using the least squares method, which has a minimum second norm distance to all points. In Fig. 15, the obtained points, the fitted lines, and the corresponding equation of each line are presented.

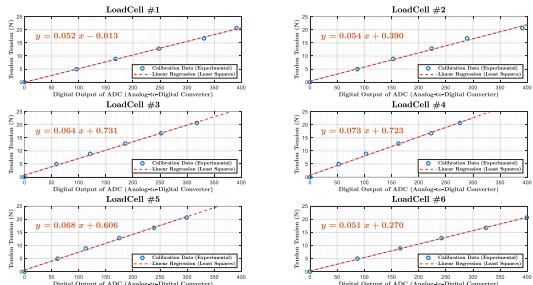


Fig. 15. Loadcells calibration results

A small amount of measurement noise remains on the output signals from the amplifier circuit of the load cells; with the design of the low-pass filter, the effect of these values will be minimized. In Fig. 16, the performance quality of the designed low-pass filter is shown in establishing a trade-off between the amount of delay and smoothing the signals.

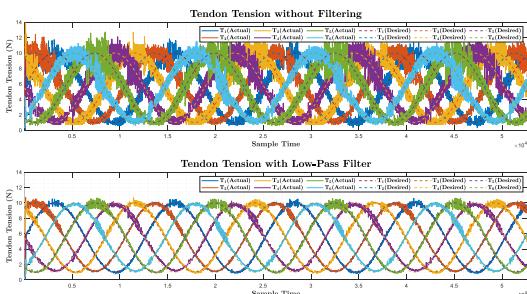


Fig. 16. Low-pass filter effect on loadcell outputs data

B. Cameras Calibration:

In order to find the momentary position of the end-effector, two cameras were used to observe the movement of the robot in the XZ and YZ planes. Fig. 17 shows the location of the cameras related to the system and their field of view.

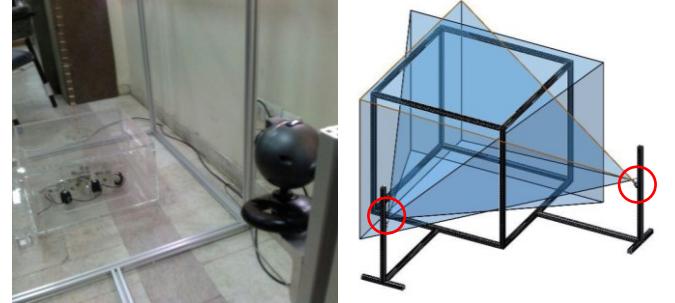


Fig. 17. Designed box for image processing system

After setting the camera filters (exposure, contrast, brightness, sharpness, etc.), the following images are provided to show their effect on the image captured by the camera. In Fig. 18, the image of the camera is presented when the filters are set (right image), and only the LED connected to the end-effector is detected by the camera as a light source. The darkening of the rest of the parts means the proper functioning of the adjusted filters.



Fig. 18. Camera original (left) and filtered (right) outputs

An important point after setting the filters related to camera calibration is to obtain the maximum error of the camera when finding the coordinates of the end-effector of the continuum robotic arm. For this purpose, the robotic arm is placed in the equilibrium position, the image processing system is allowed to run for a relatively long time, and the camera works with a frequency of 30 frames per second (the sampling time is about 33 ms).

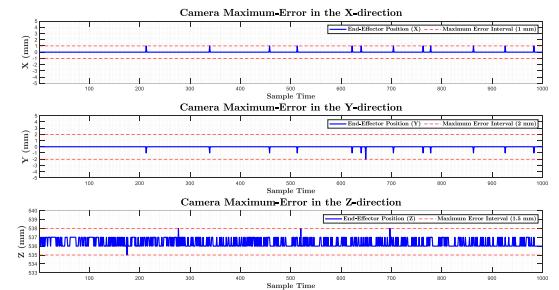


Fig. 19. Camera maximum error test

The image processing has been performed for 30 minutes. The test results are presented in Fig. 19, and according to the graphs, the maximum recorded error is equal to 2 mm, which is an acceptable value.

V. OBTAINED RESULTS

Trajectory planning can be an efficient tool to measure the performance quality of the control system. Here, the reference trajectory is considered as follows.

$$\begin{cases} x = [0.2 + 0.025 \cos(14t)] \sin(t) \\ y = [0.2 + 0.025 \cos(14t)] \sin(t) \cos(t) \\ z = 0.2 \operatorname{linsmf}(\sqrt{0.4^2 - x^2 - y^2}, [0.25, 0.4]) + 0.2 \end{cases} \quad (13)$$

In equation (13), *linsmf* is a linear s-shaped fuzzy membership function. In trajectory planning, it has been tried to consider different frequencies for trajectories, and the obtained path passes through different points in the work space of the system. Finally, a 3D path is obtained in the work space, which is the result of the above trajectory planning.

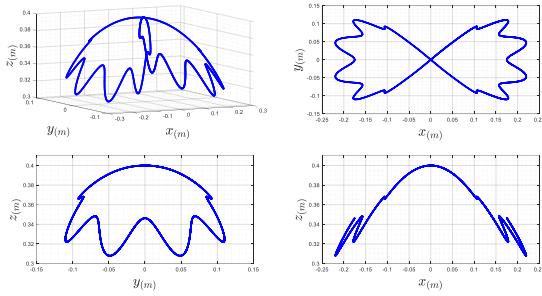


Fig. 20. Obtained 3D path for designed trajectories.

A. Simulation in MATLAB-Simulink software

To visualize the quality of trajectory tracking, an animation was created using MATLAB-Simulink and source code shared in [16]. The animation is shown in Fig. 21.

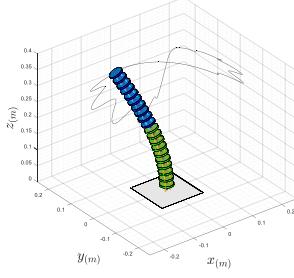


Fig. 21. Creating animation by MATLAB-Simulink

1) The 1st Scenario: Ideal Conditions (without noise and disturbance)

Simulation results of the described control systems for tracking the trajectory using the analytical model are shown below. Fig. 22 presents the controller gains set by the fuzzy inference system online during tracking. The most notable observation is the difference in the scale of the gains, which are much lower for the MTJ controller than for the TJ controller (because the correcting term with feedback linearization estimation performs most of the control). This makes the MTJ controller less sensitive to noise.

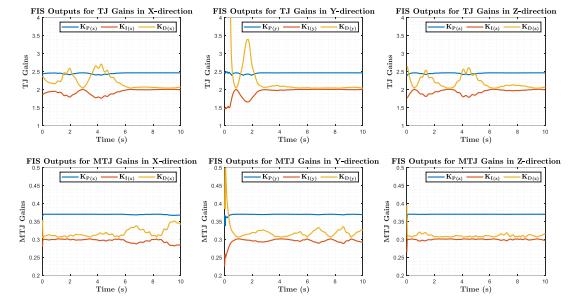


Fig. 22. TJ & MTJ gains variations in the ideal condition

Fig. 23 shows the tracking quality of the TJ and MTJ controllers with fuzzy inference system-adjusted gains. To better understand their performance, the root mean square error (RMSE) was calculated for each coordinate during the simulation, as shown in the images. The last graph shows these changes for both strategies, and its value for the error vector over time is presented.

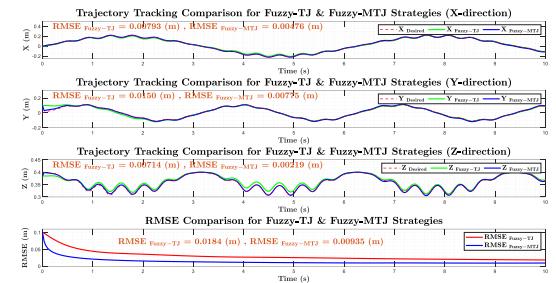


Fig. 23. Trajectory tracking quality in the ideal condition

Fig. 24 shows the diagram of tendon tension for both strategies. The tension forces are similar in magnitude, which indicates that the MTJ algorithm effectively adjusts the control input to minimize the error.

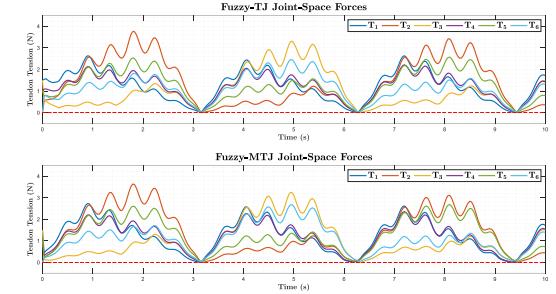


Fig. 24. Tendon tension variations in the ideal condition

2) The 2nd Scenario: Realistic Conditions (white noise & 15% disturbance)

To evaluate the trajectory tracking performance of controllers designed with gain tuning by the fuzzy inference system under realistic conditions, disturbances (unwanted input signals) and noise (unwanted output signals) are considered during the simulation. Disturbances are simulated by randomly altering the control input in joint space by 15%. White noise with an average of zero and a standard deviation of approximately 5 millimeters is used to simulate noise. The controller gains set by the fuzzy inference system online during tracking are shown in Fig. 25. The most notable

observation is that the MTJ controller gains are much lower than the TJ controller gains, which is due to the fact that the MTJ controller uses feedback linearization to perform most of the control.

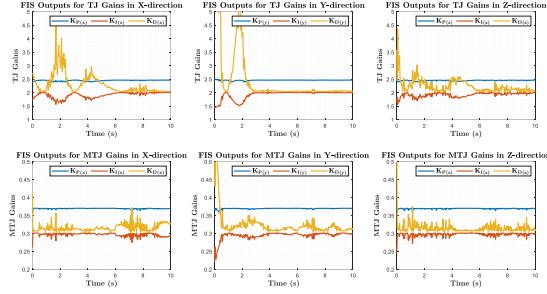


Fig. 25. TJ & MTJ gain variations in the presence of noise and disturbance

Fig. 26 shows the tracking quality of the TJ and MTJ controllers and their RMSE in the presence of noise and disturbance.

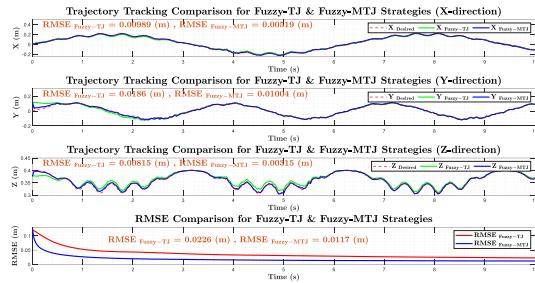


Fig. 26. Trajectory tracking quality in the presence of noise and disturbance

Fig. 27 shows the diagram of the joint space forces for the TJ and MTJ controllers in the presence of noise and disturbance. The forces are similar in magnitude, indicating that the MTJ algorithm effectively adjusts the control inputs. The TJ algorithm is more sensitive to noise and disturbance, as evidenced by the strong effect of noise and disturbance on the input signals and the loosening of the tendons in some cases. The MTJ algorithm is less sensitive to noise and disturbance because the correction term performs most of the control by estimating the feedback linearization and the MTJ algorithm has smaller control gains.

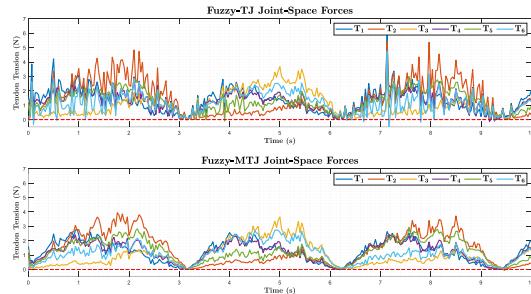


Fig. 27. Tendon tension variations in the presence of noise and disturbance

B. Implementation on a physical platform

An image of the tendon-driven continuum robotic arm developed in the ARAS laboratory is shown in Fig. 28.

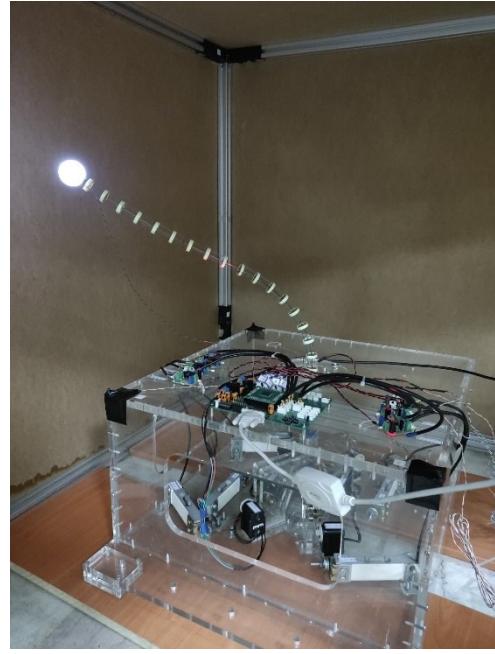


Fig. 28. The TDCR developed in the ARAS Laboratory

To verify the obtained results from simulations, a dynamic (kinetic) control strategy is implemented in a cascade structure with an inner loop to adjust the tension force of the tendons. The inner-loop controller (Decentralized PI) uses loadcell feedback to calculate the tendons' tension force at every time step and compares it to the desired tension force. The outer loop controller (Fuzzy-MTJ) works with camera output data to adjust the instantaneous position coordinates of the end effector at every moment. Fig. 29 shows the cascaded control structure customized for continuum robotic arms.

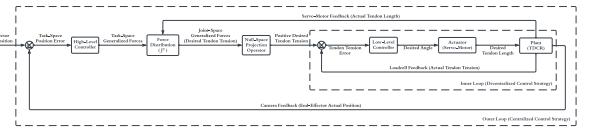


Fig. 29. Proposed cascaded control structure for closed-loop dynamic control

In the following, the results of implementing the Fuzzy-MTJ strategy on the real platform are presented to evaluate the quality of controllers designed with gain tuning by the FIS.

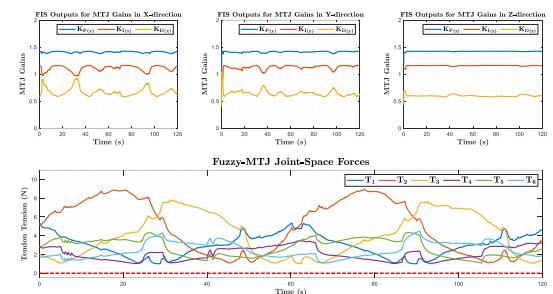


Fig. 30. MTJ gains and joint space forces in the experimental test

Fig. 30 shows the obtained gains of the controller by the FIS while tracking the trajectory (online) and the joint space forces. Fig. 31 shows the path tracking quality for the Fuzzy-MTJ controller with the RMSE variations.

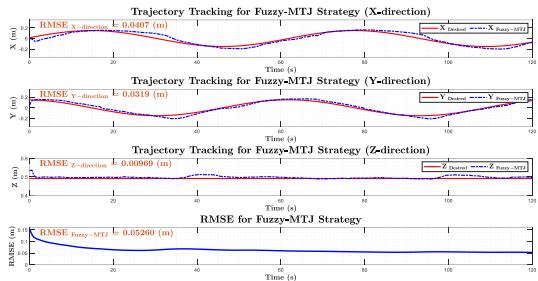


Fig. 31. Trajectory tracking quality in the experimental test

VI. CONCLUSIONS

In this paper, a new control strategy for continuum manipulators has been proposed. This approach is based on the structure of the MTJ algorithm and FIS as an intelligent gain adjustment system. To show the performance quality of designed controllers, simulations were performed in both ideal and realistic (in the presence of noise and disturbance) conditions, and the results were experimentally verified on a physical platform. The results showed that the adjusted MTJ strategy was able to track the desired path successfully with an RMSE of 1 cm in simulation and 5 cm in experimental tests. The tracking quality in the implementation on the physical system can be improved by more precisely setting the intervals for the controller gains.

REFERENCES

- [1] M. T. Chikhaoui and J. Burgner-Kahrs, "Control of Continuum Robots for Medical Applications: State of the Art," in *ACTUATOR 2018; 16th International Conference on New Actuators*, 2018, pp. 1–11.
- [2] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum Robots for Medical Applications: A Survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, Dec. 2015, doi: 10.1109/TRO.2015.2489500.
- [3] W. Amehri, "Workspace Estimation and Design Optimization of Soft Robots," Feb. 2022, Accessed: Sep. 01, 2023. [Online]. Available: <https://theses.hal.science/tel-03995027>
- [4] J. J. Craig, "Introduction to robotics : mechanics and control," p. 438.
- [5] S. A. A. Moosavian and E. Papadopoulos, "Modified transpose Jacobian control of robotic systems," *Automatica*, vol. 43, no. 7, pp. 1226–1233, Jul. 2007, doi: 10.1016/J.AUTOMATICA.2006.12.029.
- [6] S. A. A. Moosavian and E. Papadopoulos, "Control of space free-flyers using the modified transpose Jacobian algorithm," *IEEE International Conference on Intelligent Robots and Systems*, vol. 3, pp. 1500–1505, 1997, doi: 10.1109/IROS.1997.656557.
- [7] F. Piltan, M. Eram, M. Taghavi, O. R. Sadnia, and M. Jafari, "Nonlinear Fuzzy Model-base Technique to Compensate Highly Nonlinear Continuum Robot Manipulator," *International Journal of Intelligent Systems and Applications*, vol. 5, no. 12, pp. 135–148, Nov. 2013, doi: 10.5815/IJISA.2013.12.12.
- [8] P. Qi, C. Liu, L. Zhang, S. Wang, H.-K. Lam, and K. Althoefer, "Fuzzy logic control of a continuum manipulator for surgical applications," 2014 *IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, 2014, Accessed: Aug. 19, 2023. [Online]. Available: https://www.academia.edu/12133765/Fuzzy_Logic_Control_of_a_Continuum_Manipulator_for_Surgical_Applications
- [9] W. Ba, X. Dong, A. Mohammad, M. Wang, D. Axinte, and A. Norton, "Design and Validation of a Novel Fuzzy-Logic-Based Static Feedback Controller for Tendon-Driven Continuum Robots," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 6, pp. 3010–3021, Dec. 2021, doi: 10.1109/TMECH.2021.3050263.
- [10] M. Goharimanesh, A. Mehrkish, and F. Janabi-Sharifi, "A Fuzzy Reinforcement Learning Approach for Continuum Robot Control," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 100, no. 3–4, pp. 809–826, Dec. 2020, doi: 10.1007/S10846-020-01237-6/METRICS.
- [11] P. Qi, C. Liu, A. Ataka, H. K. Lam, and K. Althoefer, "Kinematic Control of Continuum Manipulators Using a Fuzzy-Model-Based Approach," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5022–5035, Aug. 2016, doi: 10.1109/TIE.2016.2554078.
- [12] C. Song, G. Gao, P. Wang, and H. Wang, "Kinematics and Fuzzy Control of Continuum Robot Based on Semi-closed Loop System," 2022 2nd International Conference on Robotics, Automation and Artificial Intelligence, RAAI 2022, pp. 43–51, 2022, doi: 10.1109/RAAI5146.2022.10092975.
- [13] X. Wang, Y. Li, and K. W. Kwok, "A Survey for Machine Learning-Based Control of Continuum Robots," *Front Robot AI*, vol. 8, p. 730330, Sep. 2021, doi: 10.3389/FROBT.2021.730330/BIBTEX.
- [14] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control Strategies for Soft Robotic Manipulators: A Survey," *Soft Robot*, vol. 5, no. 2, pp. 149–163, Apr. 2018, doi: 10.1089/SORO.2017.0007/ASSET/IMAGES/LARGE/FIGURE12.JPG.
- [15] H. Yuan, L. Zhou, and W. Xu, "A comprehensive static model of cable-driven multi-section continuum robots considering friction effect," *Mech Mach Theory*, vol. 135, pp. 130–149, May 2019, doi: 10.1016/J.MECHMACHTHEORY.2019.02.005.
- [16] P. Rao, Q. Peyron, S. Lilge, and J. Burgner-Kahrs, "How to Model Tendon-Driven Continuum Robots and Benchmark Modelling Performance," *Front Robot AI*, vol. 7, p. 630245, Feb. 2021, doi: 10.3389/FROBT.2020.630245/BIBTEX.
- [17] B. A. Jones and I. D. Walker, "Kinematics for multisection continuum robots," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 43–55, 2006, doi: 10.1109/TRO.2005.861458.
- [18] I. S. Godage, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "Path planning for multisection continuum arms," 2012 *IEEE International Conference on Mechatronics and Automation, ICMA 2012*, pp. 1208–1213, 2012, doi: 10.1109/ICMA.2012.6283423.
- [19] Z. Y. Zhao, M. Tomizuka, and S. Isaka, "Fuzzy Gain Scheduling of PID Controllers," *IEEE Trans Syst Man Cybern*, vol. 23, no. 5, pp. 1392–1398, 1993, doi: 10.1109/21.260670.
- [20] K. L. Anderson, G. L. Blankenship, and L. G. Lebow, "Rule-based adaptive PID controller," *Proceedings of the IEEE Conference on Decision and Control*, pp. 564–569, Dec. 1988, doi: 10.1109/CDC.1988.194374.