Research Article

Modelling And Control of Cable Driven Robotic Arm Using Maplesim

Ahmed ALKAMACHI[1,*], *Yahya Ghufran Khidhir ABBOOSH*[2]

[1]University of Baghdad, Al-Khwarizmi College of Engineering, Mechatronics Engineering Department, Al-Jadriya, Baghdad, Iraq
[2]Northern Technical University, Technical Engineering College/Kirkuk, Electronic and Control Engineering Department, Kirkuk, Iraq

ahmed78@kecbu.uobaghdad.edu.iq, yahhya.khidhir24@ntu.edu.iq

*Corresponding author: Ahmed ALKAMACHI; ahmed78@kecbu.uobaghdad.edu.iq

DOI: 10.15598/aeee.v22i3.5685

Article history: Received Jan 12, 2024; Revised Mar 14, 2024; Accepted Apr 22, 2024; Published Sep 30, 2024.

Abstract. *Cable-driven robotic arms (CDRA) are robots with novel structures, wherein flexible cables are used to drive rigid links identified to move the end effector according to a desired trajectory. Due to the complex and nonlinear characteristics of this type of robotic arm, it is challenging to derive the model, which requires critical analysis to be conducted. This paper presents the design, modeling, and Model Predictive Control (MPC) of a special 2D CDRA with four rigid links. Maplesim is employed as a tool to design and simulate the proposed robotic arm. First, the prototype model is constructed in Maplesim and simulated using random input signals, and the input and output data sets are collected. A data-driven scheme based on neural networks is used to learn the unknown kinematics of the CDRA and to solve the kinematic control issue. The Matlab-Simulink platform is used for this purpose, and the black box model is obtained using the neural network fitting tool. MPC is then used for the end effector trajectory tracking control and to validate the modeling processes. Furthermore, comparative simulations using two scenarios are applied to the controlled system to verify the effectiveness of the proposed modeling and control method with the aid of Mean Squared Error (MSE) as an optimality index. The result verified that CDRA is capable of following reference trajectories accurately with MSE of 10e-5 and 4.99e-5 for rhombus and circular trajectories respectively.*

Keywords

Model Predictive Control, Maplesim, Cable-driven robotic arms, Data-driven, Neural Networks.

1. Introduction

Robotic manipulators are considered one of the most significant robotic applications since they qualify the robots to interact with their surrounding. [1]. In recent years, cable-driven robots have gained lots of interests due to their individual features, such as safe human–robot interactions, flexibility, high load-to-weight ratio [2]. Using cables to drive the manipulators provide a compact and significantly small inertia manipulator since the actuators are placed at its base [3]. Bioinspired by the structure of humanoid arms, a cable driven robotic arm (CDRA) is introduced for safe human–robot interaction, which utilizes cables to simulate the functionality of the humanoid muscles. In CDRAs, rigid links are changed with cables which have great advantages in reducing the mass and inertia of robotic arms and elimination of revolute joints. These advantages permit the end effector to reach high motion acceleration in considerable workspaces [4] which will make them suitable in many applications [5], especially in pick and place processes [6]. These types of manipulators can also be used for Aerial construction and inspection purposes by attaching them to a quadcopter [7]. In the last decades, different model of CDRAs have been designed for various applications, such as robots designed for service [8], high speed manipulation [9], medical application [10, 11], rehabilitation tasks [9, 12–14], adaptive grasping robotic hand [15], inspection processes [16, 17], and for haptic applications [18]. For almost all CDRAs, driving cables are traced through the linkages from nearer actuators to their corresponding remote joints. Due to the high coupling effects of multiple degrees of freedom, the kinematic model becomes more complicated and difficult to derive mathematically, which may also be affected by their nonlinear dynamic behaviors, cable hysteresis characteristics, structural design, the way the cable routed, the size and the geometrical place of pulleys, etc. [19, 20]. However, these type of robotics arms suffers from several drawbacks, such as their low accuracy, high vibration, etc. [21] that limits their application areas. For the aforementioned reasons, trajectory tracking is of great importance for CDRAs to achieve various tasks. However, it is challenging for CDRAs to follow the trajectory in demand smoothly and precisely due to their nonlinear characteristics and parameters variation.In [22], the author used solidworks and Simscape platform to model and control a dynamical system without the need of explicit mathematical equations. From the literatures, the mechanics of cable driven robot is quite complicated because of the high nonlinearities and continuous deformations of cables. Unlike traditional robotics arm, the standard robot modeling tools can not be applied to the cable driven manipulators [23]. Plentiful attempts have been made to model the complex kinematics and dynamics of the cable driven robots. Optimization methods are used in [24, 25] to analyze the inverse dynamic problem, while [26, 27] use the heuristic-based methods for the same issue. A Neural network-based modeling has been used to obtain a black box for a continuum robot [28]. The accuracy of the obtained model is limited to the experimental situation such as external payloads and training data. The establishment of a relation between the actuator space to joint space is necessary. Uyguroglu ˘ and Demirel used oriented graph approach for the kinematic analysis of cable-driven robotic mechanisms [29].

In this work, we aim to accurately obtain the dynamics of a proposed CDRA system by capturing the actual response of the model taking the effect of wire elasticity, internal friction of the system parts, material damping, as well as the hysteresis. Maplesim is used to simulate three joints – three pullies CDRA and then the input – output data pairs are collected. These data are transferred to Matlab to train a black box neural network which will represent the derived model. One important goal of our approach is to establish a method usable to model any cable-driven systems and improve their position tracking accuracy. moreover, our method is eligible to sense the magnitude and location of the force, torque, and wire tensions. Such data can help in control the grasping force and cable characteristics requirements.

The remaining of the paper are organized as follows: In Section 2. , our proposed CDRA system is introduced and the Maplesim modeling is explained and analyzed. Based on the obtained system model, MPC control scheme is designed and simulations are conducted in Section 3. . The results are also discussed in Sections 3. , and finally Section 4. conclude the paper by showing a summary of the main contributions.

2. System Design and Modeling

2.1. System Setup

For the vast majority of traditional robotic manipulators, each kinematic joint is driven directly by individual actuator placed at the joint link. Clearly, it can be notice that each actuator directs one joint independently, which means that there is no coupling between the joint and actuator pairs. Therefore, the relationship between actuator space and joint space can be easily derived. However, the exceptional characteristic of a CDRA structure is that all the actuators are positioned at the base of the manipulator far away from the joints. The cable pulls movements track the routing from the ends of the joints between the actuators and the controlled joints. System modeling is based on the CDRA platform shown in Fig. 1, which consists of four rigid links, three revolving joints, and three pulleys which are driven by stepper motors to actuate three cables. As it can be seen from Fig. 1, cable A which is connected to link A, passed through link C and B and it is pulled by pulley A. pulley B control the pulling of cable B which passed through link C and connected to link B while link C is controlled by pulley C through cable C. The main parameters for the robotic manipulator parts are listed in Table 1.

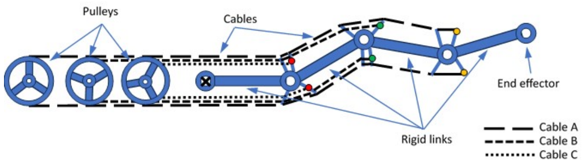

Fig. 1: CDRA platform drawing.

Tab. 1: Properties of CDRA parts.

Parts	Properties
Rigid link (link A to link D)	Dimensions: 60X60X400 mm Mass for each link: 100 g
Pully (pulley A to pulley C)	Radius: 100 mm, Rotation range: – pi to pi, Mass for each pulley: 100g
Cable	Radius: 4mm
Revolving joint	Constrains: – pi/3 to pi/3
Elasto-Backlash	Spring constant: 100000 N.m/rad, Damping constant: 1000 N.m.s/rad, Total Backlash: pi/3 rad

2.2. Maplesim Model

Maplesim is an advance modeling, simulation, and analysis tools that simulate systems at symbolic level and allows researcher to use graphical tools to apply modern techniques and develops and analyze design models instead of mathematical equations. This software tool (Maplesim 2023) is used in this work to build the model of the cable driven robot manipulator as shown in Fig. 2. The 2D Maplesim symbolic model is realized from the free body drawing in Fig. 1 using the parameters in Table 1. It consists of five main components, namely pulleys, ropes, CAD geometry with rigid body, elasto-backlash, and end effector position sensor. To provide smooth motion for a cable-driven robot, elasto-backlash components with predetermined stiffness are used. Elasto-backlashs are applied at the joints in simulations to improve realism and accuracy.

In order for the model to be consistent with different types of reference inputs, chirp components are used to provide randomized signals for the system. Their amplitudes range from pi to -pi and their instantaneous frequencies increase or decrease linearly over time. The output which is the end effector position is measured using a position sensor. The input – output pairs will be used to obtain the system black box using neural networks as it will be shown in the next subsection.

2.3. Neural Network Model

With the rapid development of artificial intelligence, neural network fitting procedures have opened up a new direction for robot modeling. This type of modeling uses real input and output data to train a network and evaluate its progress using mean square error. This measurement can help quantify how successfully the black box model captures system dynamics. Three motor-driven pulleys are linked to linear chirp signals that generate chirps with a frequency that varies linearly with time in order to apply variable pulleys' angles to the cables and actuate the linkages and joints, respectively. A neural network based black box models of the cable-driven robot can be obtained directly from the data-driven inputs and outputs of the system, without the need for a precise mathematical model of the system, thus avoiding the modeling and parameter tuning process that relies on expert experience. In order to get the required input – output data, the robotic arm is simulated under the advanced Maplesim environment. The chirp inputs are applied and the end effector position is recorded. Furthermore, the input – output data sets are exported to Excel sheet to be used in Matlab Neural Net fitting tool. Fig. 3 and Fig. 4 represent the input signals and output position plots.

n neural network fitting, feedforward neural network equation has been used. The input data received by the neuron multiplied by corresponding weights and summed up by a bias term as in Eq. (1) below:

$$z = \sum_{i=1}^{n} (w_i \times x_i) + b, \tag{1}$$

where z is the total input weights and bias, w_i indicates the weights corresponding to each input x_i, x_i represents the input variables, b is the bias, n is the total number of inputs.

Then the hyperbolic tangent function activation function is applied in order to add nonlinearity and calculate the output of the neuron [30]. By training multiple layers, the neural network learns complicated relationships between inputs and outputs, with weights and biases adjusted to reduce prediction error using mean square error.

The recorded data from Maplesim are used in the Neural Fitting tool (nftool) where the data should be divided into three portions for training, validation, and test intents. The tool also authorizes the user to nominate the number of hidden neurons. In order to reach the best settings for the tool, several experiments were done as follows:

1. Splitting data as follows: 80% training, 10% validation, and 10% testing while the number of neurons is 20.
2. Splitting data as follows: 70% training, 15% validation, and 15% testing, while the number of neurons is 10.
3. Splitting data as follows: 70% training, 15% validation, and 15% testing, while the number of neurons is 20.

The networks are trained to fit the outputs to the targets using Levenberg – Marquardt algorithm and the training results are tabulated in Table 2 where MSE is the Mean Squared Error between the output and the target, and R is the Regression value that measure the correlation between the output and the target.

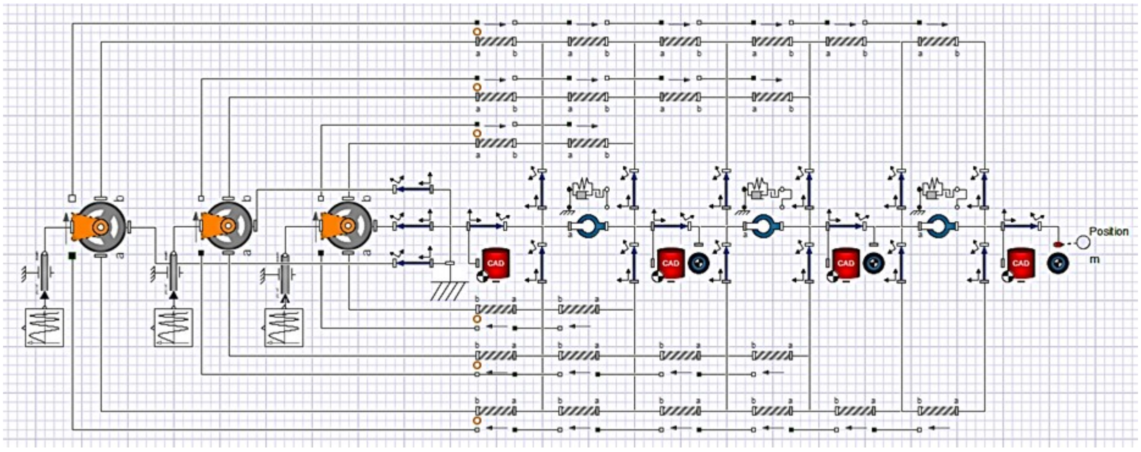

Fig. 2: CDRA platform Maplesim composition.

Tab. 2: Neural network training results.

Exp.	Division %	No. of Neurons	MSE ± Std	R	CIs
1	Training 70%	10	1.77553e-3 ± 7.9617e-04	0.998903	0.997807
	Validation 15%		1.82997e-3 ±0.0017	0.998893	0.997787
	Testing 15%		1.78486e-3 ± 0.0017	0.998896	0.997793
2	Training80%	20	4.56565e-4 ± 3.7767e-04	0.999720	0.999440
	Validation 10%		4.23544e-4 ±0.0010	0.999728	0.999456
	Testing 10%		6.12947e-4 ± 0.0012	0.999620	0.999240
3	Training 70%	20	2.59517e-4 ±3.0439e-04	0.999840	0.999680
	Validation 15%		2.49828e-4 ±6.4528e-04	0.999848	0.999696
	Testing 15%		3.00858e-4 ±7.0812e-04	0.999810	0.999620

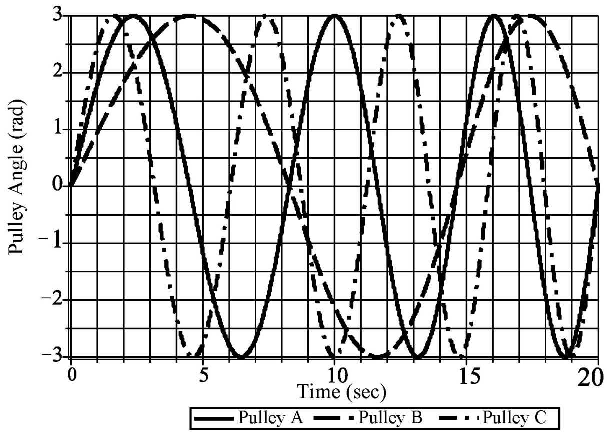

Fig. 3: Input signals.

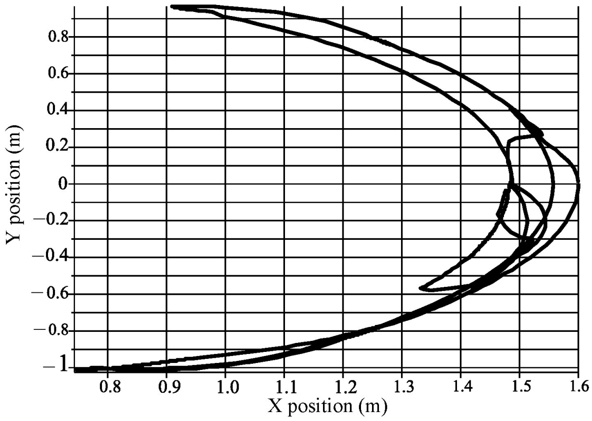

Fig. 4: Output position plots.

From the results, it can be shown that the best setting is splitting data as follows: 70% training, 15% validation, 15% testing, and the number of neurons is 20. Increasing number of neurons will improve the regression results but at a cost of increasing the computation time which is not worth since the regression value for this setting shows that the obtained model is very close to the actual one.

The neural network layers configuration is shown if Fig. 5 where twenty hidden neurons are used through the process.

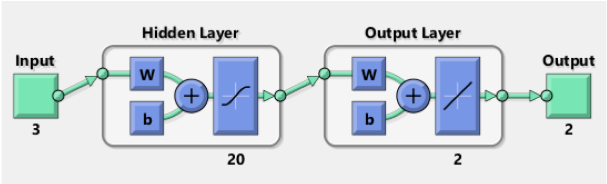

Fig. 5: Neural network layers configuration.

The error histogram in Fig. 6 visualizes the error between the target and predicted output. Since the error plot is distributed to zero, it means that the model performs well. The regression plots in Fig. 7 show that the regression values are very close to 1 in training, validation, test, and in the overall performance. It means that the model is fitted well according to the data collected from the Maplesim model.

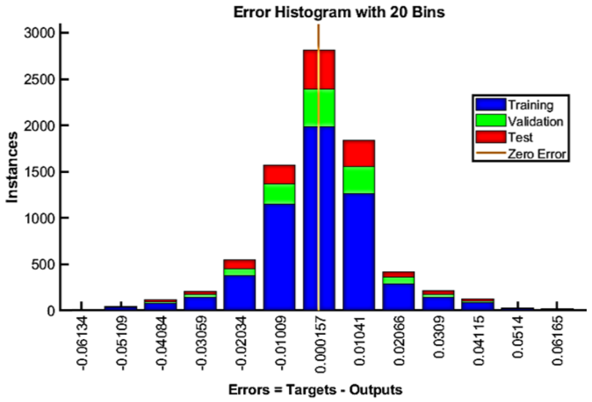

Fig. 6: Error histogram for the model fitting process.

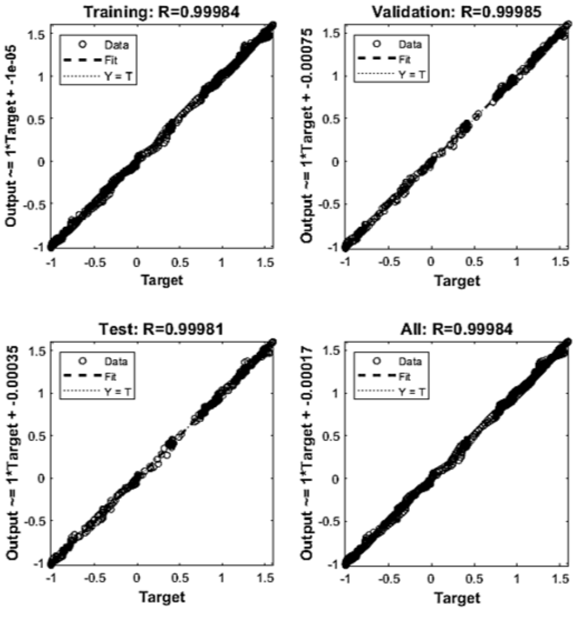

Fig. 7: Regression plots for the training, validation, test, and in the overall performance.

2.4. Controller Design

Due to the high-dimensional continuous state space and nonlinear characteristics of the cable-driven robot, MPC is preferred over other traditional control methods. It is successful at handling complicated systems with various inputs, outputs, and limitations. MPC employs the system's predictive model to make future predictions and optimize control operations accordingly, allowing for improved performance even in the presence of disturbances. MPC can handle constraints on the system's inputs, outputs, and states, guaranteeing that they are not violated during control action. Changes in system dynamics or operating conditions can be easily incorporated into MPC by updating the predictive model and optimizing control actions based on the updated data. In addition, MPC optimizes a cost function that takes into account both control objectives and constraints [31].

A neural network models the cable-driven robot directly from the data-driven inputs and outputs of the system without the need for derivation of a precise mathematical equations of the system. In this paper, we consider a scenario where the dynamics of a nonlinear system are unknown. However, the input and output data are available. A fitted model is learned from these data using a neural network. The model in turn be controlled using MPC scheme. The selected controller is an optimal controller that is used when the model of the system being controlled is available. It is required to control the position of the robotic arm end effector through predefined trajectories. MPC relies on a model of the plant to predict future trajectories and to optimize over them. It can also handle constraints which are important, as violating them may lead to undesired consequences. The goal of MPC is to minimize a predefined cost function while satisfying constraints such as system dynamics, actuator limitations, etc [32]. At each time step, it calculates the best set of control actions (trajectories) that minimizes the cost function over a specific time horizon and pick the action for the most immediate time step and the process repeats at the following time step [33]. The overall system Simulink representation is shown in Fig. 8.

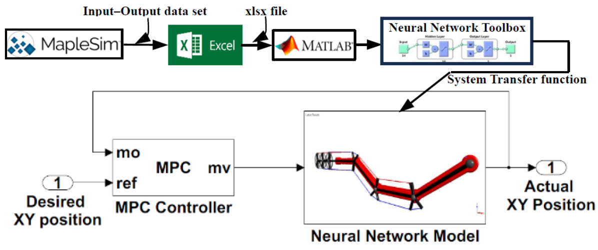

Fig. 8: System Simulink block diagram.

The quadratic cost function for optimization is given by:

$$J = \sum_{i=1}^{N} \omega_{x_i} (r_i - x_i)^2 + \sum_{i=1}^{M} \omega_{u_i} \Delta u_i^2, \qquad (2)$$

where x_i is the controlled variables (measured outputs of the model XY coordinates of end effector), r_i is the reference variables (desired outputs: desired XY coordinates of end effector), u_i manipulated variables (Model's inputs: pulleys' angles) ω_{x_i} weighting coefficient reflecting the relative importance of x_i, ω_{u_i} weighting coefficient reflecting the relative importance of u_i. In MPC, the control horizon indicates how long the system will be subjected to a control sequence. A larger control horizon improves control performance, but it may also result in more aggressive control actions [34]; therefore, we chose a control horizon equal to 10 as shown in Table 3 by using the trial-and-error approach and simulation-based analysis.

Tab. 3: Properties of CDRA parts.

Parameter	Value
Sample time	0.005 s
Prediction horizon	60
Control horizon	10
Input Constraints	Min: $-\pi/3$, Max: $\pi/3$, RateMin: $-\pi/16$, RateMax: $\pi/16$
Output Constraints	Min: -1.6, Max:1.6
Input Weights	Weight: 0.0001, Rate Weight: 5e-05, Target: nominal
Output Weights	Weight: 1

The interval at which the controller estimates new control actions is known as the sampling time. While a longer sample interval may result in slower system functioning, a smaller sampling time offers faster responsiveness but increases computing effort [33]. Because our work is simulation-based, we used a short sampling time of 0.005 seconds in order to achieve the fastest responsiveness.

The link between performance and computing complexity is mostly balanced by the prediction horizon. A longer prediction horizon means more future information is taken into account, which improves performance and requires more computing. Conversely, a shorter prediction horizon leads to a reduction in computing complexity but a corresponding drop in performance [33]. A high prediction horizon of 60 has been used, as indicated in Table 3, to achieve high performance.

The physical input constraints that must be obeyed are actuator limitations like maximum allowable angle horizon of each joint, in order to bypass the singularity and are taken as $-\pi/3$ to $\pi/3$. In the other hand, the output constrains represent the allowable work space for the end effector. Since the primary goal of the manipulator with the controller is to track the reference XY position of the end effector, the weights of outputs (ω_{x1} and ω_{x2} for X and Y coordinates respectively) in Eq. (2) are set to one as mentioned in Table 3.

3. Simulation and Path Tracking Scenarios

In this section, in order to illustrate the validity of the MPC-controller in term of position control of CDRA as well as the accuracy of the proposed modeling method, two experiments are carried out to determine the position difference error along X and Y coordinates with and without MPC controller. Our aim is to track the reference trajectory with less displacement error. Linear and circular reference trajectories were designed in these experiments. The Mean Square Error (MSE) is used as a performance index to examine the system and controller ability to track a desired trajectory [35]. MSE calculates the squared difference between two trajectories as shown in the equation below:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(r_i - \hat{x}_i)^2, \tag{3}$$

where r_i represents the reference trajectory of points while $\hat{x}_i$ is the simulation output points. n is the number of trajectory data points.

3.1. First Scenario

The first reference trajectory is combined of four linear motions to shape a rhombus trajectory. The end-effector starts from rest position where X and Y coordinates is (1.6,0) m respectively. Then it travels through (1.4,0.4), (1.2,0), (1.4,-0.4) m and then it returns back to the initial point at (1.6,0) m after moving through the desired trajectories as shown in Fig. 9.

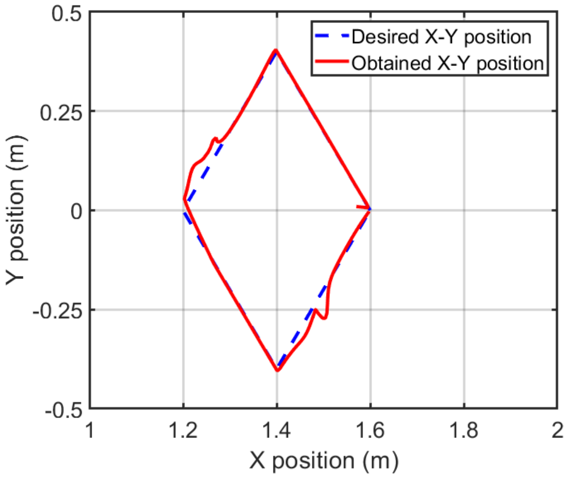

Fig. 9: Rhombus trajectory tracking.

The MSE error for the above trajectory is 10e-5 which is very small for such metric sized robot manipulator. From Fig. 10, it can be noticed that the MPC output trajectory has been tracked the rhombus reference trajectory stated above in 20 seconds and maximum error occurs at 8.18 and 17.35 sec. Reference to Fig. 11, it can be noticed that the peak errors occur at the start and end of pulley C operation.

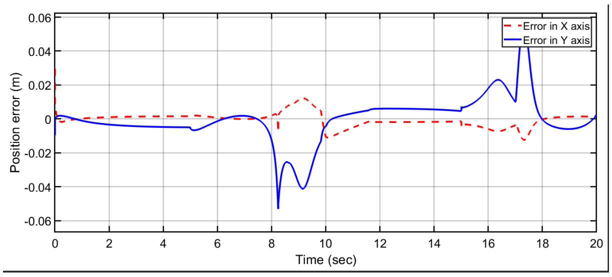

Fig. 10: Position error of rhombus trajectory tracking.

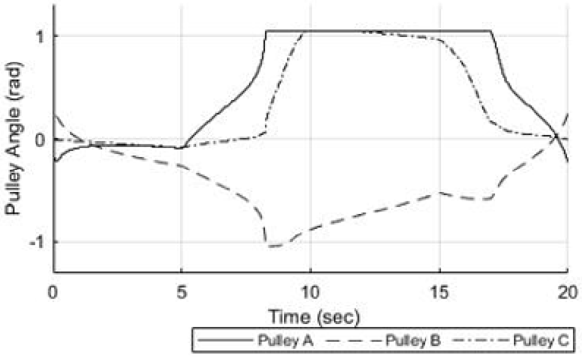

Fig. 11: Pulleys inputs to neural network plant for rhombus trajectory tracking.

3.2. Second Scenario

The second trajectory is a circular trajectory with radius of 0.15 cm and starting point of (X, Y) = (1.6,0). A circular motion of 360-degree counterclockwise occurs in X-Y plane, and the end-effector returns to the prior position of (X, Y) = (1.6,0) Fig. 12.

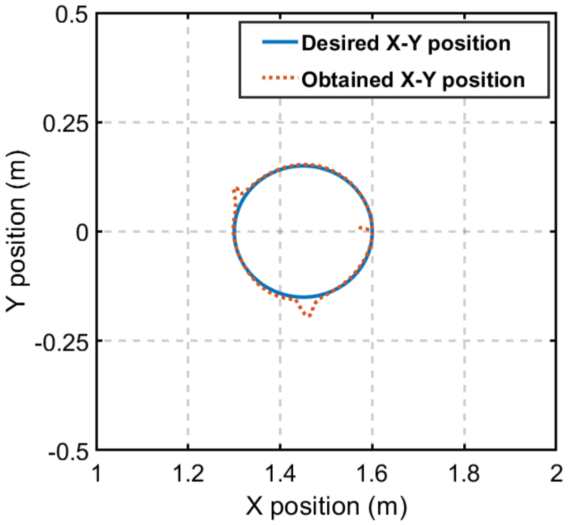

Fig. 12: Circular trajectory tracking.

From the calculated results, it is obtained that the circular trajectory has less MSE of 4.99e-05.

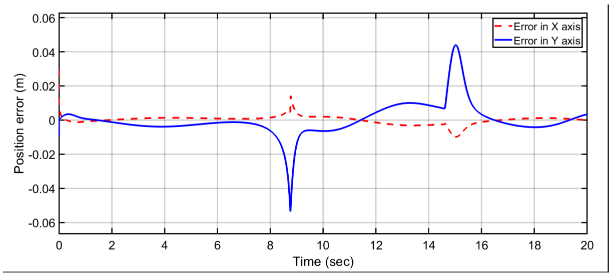

Fig. 13: Position error of circular trajectory tracking.

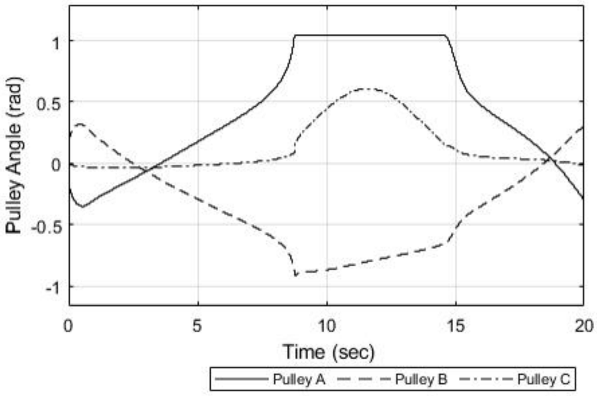

Fig. 14: Pulleys inputs to neural network plant for circular trajectory tracking.

In this scenario, according to Fig. 13 the MPC controller made the manipulator model to track the reference circular trajectory with the peak errors occur at 8.75 and 15.1 sec, which is again as a result of the start and end point of pulley C engagement as shown in Fig. 14. The result in both linear and circular reference are ideal in term of tracking error values.

4. Conclusion

In conclusion, this study introduces a 2D CDRM integrated with an MPC for precise position control. Linear and circular trajectories are utilized to prove the effectiveness of the modeling procedure and the MPC controller. By strategically assigning weights to critical output elements, namely the end-effector's linear and angular positions, the proposed design offers a straightforward structure devoid of complex mathematical equations. The results are promising, affirming the efficiency of the proposed modeling method and the obtained model with the MPC controller in tracking both linear and angular trajectories within acceptable time and error constraints. Looking ahead, there is a promising avenue for future research to implement this data-driven CDRM in real-time applications, further advancing the field of robotic manipulation.

Author Contributions

ALKAMACHI and ABBOOSH developed the theoretical formalism, performed the analytic calculations and performed the numerical simulations. Both ALKAMACHI and ABBOOSH contributed to the final version of the manuscript. ALKAMACHI supervised the project.

References

[1] DAL VERME L. D. M. C., D. LUDOVICO, A. PISTONE, C. CANALI, and D. G. CALDWELL. Lyapunov stability of cable-driven manipulators with synthetic fibre cables regulated by non-linear full-state feedback controller. *ISA transactions*. 2023, vol. 142, pp. 360-371. DOI: j.isatra.2023.08.019.

[2] WANG, Y., L. LIU, J. CHEN, F. JU, B. CHEN, and H. WU. Practical robust control of cable-driven robots with feedforward compensation. *Advances in Engineering Software*. 2020, vol. 145, p. 102801. DOI: j.advengsoft.2020.102801.

[3] DHYAN, S. B., C. C. HOANG, B. S. HAN, J. Y. TAN, W. T. CHOW et al.. Mitigate inertia for wrist and forearm towards safe interaction in 5-dof cable-driven robot arm. In: *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. Seattle, WA, USA: IEEE, 2023, pp. 215-220. DOI: 10.1109/AIM46323.2023.10196261.

[4] KAWAMURA, S., H. KINO, and C. WON. High-speed manipulation by using parallel wire-driven robots. *Robotica*. 2000, vol. 18, iss. 1, pp. 13-21. DOI: 10.1017/S0263574799002477.

[5] ALP, A. B. and S. K. AGRAWAL. Cable suspended robots: design, planning and control. In: *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292)*. Washington, DC, USA: IEEE, vol. 4. 2002, pp. 4275-4280. DOI: 10.1109/ROBOT.2002.1014428.

[6] LIU. P., H. TIAN, X. CAO, X. QIAO, L. GONG, X. DUAN, Y. QIU, and Y. SU. Pick-and-place trajectory planning and robust adaptive fuzzy tracking control for cable–based gangue-sorting robots with model uncertainties and external disturbances. *Machines*. 2022, vol. 10, iss. 8, p. 714. DOI: 10.3390/machines10080714.

[7] ALAKHRAS, A., I. H. SATTAR, M. ALVI, M. W. QANBAR, M. A. JARADAT, and M. ALKADDOUR. The design of a lightweight cable aerial manipulator with a cog compensation mechanism for construction inspection purposes. *Applied Sciences*. 2022, vol. 12, iss. 3, p. 1173. DOI: 10.3390/app12031173.

[8] JIANQING P., W. HAOXUAN, L. TIANLIANG, and H. YU. Workspace, stiffness analysis and design optimization of coupled active-passive multilink cable-driven space robots for on-orbit services. *Chinese Journal of Aeronautics*. 2023, vol. 36, iss. 2, pp. 402-416. DOI: 10.1016/j.cja.2022.03.001.

[9] HWANG, M., B. THANANJEYAN, S. PARADIS, D. SEITA, J. ICHNOWSKI, D. FER, T. LOW, and K. GOLDBERG. Efficiently calibrating cable-driven surgical robots with rgbd fiducial sensing and recurrent neural networks. *IEEE Robotics and Automation Letters*. 2020, vol. 5, iss. 4, pp. 5937-5944. DOI: 10.1109/LRA.2020.3010746.

[10] LARIBI, M. A., M. CECCARELLI, J. SANDOVAL, M. BOTTIN, and G. ROSATI. Experimental validation of light cable-driven elbow-assisting device l-cadel design. *Journal of Bionic Engineering*. 2022, vol. 19, iss. 2, pp. 416–428. DOI: 10.1007/s42235-021-00133-5.

[11] HAMIDA, I. B., M. A. LARIBI, A. MLIKA, L. ROMDHANE, S. ZEGHLOUL, and G. CARBONE. Multi-objective optimal design of a cable driven parallel robot for rehabilitation tasks. *Mechanism and Machine Theory*. 2021, vol. 156, p. 104141. DOI: 10.1016/j.mechmachtheory.2020.104141.

[12] KUAN, J.-Y., K. A. PASCH, and H. M. HERR. A high-performance cable-drive module for the development of wearable devices. *IEEE/ASME Transactions on mechatronics*. 2018, vol. 23, iss. 3, pp. 1238–1248. DOI: 10.1109/TMECH.2018.2822764.

[13] CHEN, T., R. CASAS, and P. S. LUM. An elbow exoskeleton for upper limb rehabilitation with series elastic actuator and cable-driven differential. *IEEE Transactions on Robotics*. 2019, vol. 35, no. 6, pp. 1464-1474. DOI: 10.1109/TRO.2019.2930915.

[14] XIONG, H. and X. DIAO. A review of cable-driven rehabilitation devices. *Disability and Rehabilitation: Assistive Technology*. 2020, vol. 15, iss. 8, pp. 885-897. DOI: 10.1080/17483107.2019.1629110.

[15] NIKAFROOZ, N. and A. LEONESSA. A single-actuated, cable-driven, and self-contained robotic hand designed for adaptive grasps. *robotics*. 2021, vol. 10, iss. 4, pp. 109. DOI: 10.3390/robotics10040109.

[16] DUPONT, P. E., N. SIMAAN, H. CHOSET, and C. RUCKER. Continuum robots for medical interventions. *Proceedings of the IEEE*. 2022, vol. 110, no. 7, pp. 847-870. DOI: 10.1109/JPROC.2022.3141338.

[17] WANG, M., D. PALMER, X. DONG, D. ALATORRE, D. AXINTE, and A. NORTON. Design and development of a slender dual-structure continuum robot for in-situ aeroengine repair. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid, Spain: IEEE, 2018, pp. 5648-5653. DOI: 10.1109/IROS.2018.8594142.

[18] ZHAO, B., S. ZHANG, Z. WU, B. YANG, and K. XU. Combx: Design and experimental characterizations of a haptic device for surgical teleoperation. *The International Journal of Medical Robotics and Computer Assisted Surgery*. 2020, vol. 16, iss. 1, p. e2042. DOI: 10.1002/rcs.2042.

[19] FABRITIUS, M. and A. POTT. A forward kinematic code for cable-driven parallel robots considering cable sagging and pulleys. In: *International Symposium on Advances in Robot Kinematics*. Springer, 2020, pp. 218–225. DOI: 10.1007/978-3-030-50975-0_27.

[20] CHEN, Q., Y. QIN, and G. LI. Qpso-mpc based tracking algorithm for cable-driven continuum robots. *Frontiers in Neurorobotics*. 2022, vol. 16, p. 1014163. DOI: 10.3389/fnbot.2022.1014163.

[21] MENDEZ, S. J. T. *Low mobility cable robot with application to robotic warehousing*. University of Waterloo 2014.

[22] ALKAMACHI, A. Integrated solidworks and simscape platform for the design and control of an inverted pendulum system. *Journal of Electrical Engineering*. vol. 71, iss. 2, pp. 122–126. DOI: 10.2478/jee-2020-0018.

[23] DAS, A. and M. NABI. A review on soft robotics: Modeling, control and applications in human-robot interaction. In: *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. Greater Noida, India: IEEE, 2019, pp. 306-311. DOI: 10.1109/ICCCIS48478.2019.8974461.

[24] P. R. MIERMEISTER. *Model selection and parameter optimization for cable-driven parallel robots*. 2021.

[25] LAU, D., D. OETOMO, and S. K. HALGAMUGE. Inverse dynamics of multilink cable-driven manipulators with the consideration of joint interaction forces and moments. *IEEE Transactions on Robotics*. 2015, vol. 31, iss. 2, pp. 479–488. DOI: 10.1109/TRO.2015.2394498.

[26] POTT, A. An improved force distribution algorithm for over-constrained cable-driven parallel robots. In: *Computational Kinematics: Proceedings of the 6th International Workshop on Computational Kinematics (CK2013)*. Springer, 2014, pp. 139–146. DOI: 10.1007/978-94-007-7214-4_16.

[27] JU, R., D. ZHANG, J. XU, H. YUAN, Z. MIAO, M. ZHOU, and Z. CAO. Design, modeling, and kinematics analysis of a modular cable-driven manipulator. *Journal of Mechanisms and Robotics*. 2022, vol. 14, iss. 6, p. 060903. DOI: 10.1115/1.4054206.

[28] GIORELLI M., F. RENDA, G. FERRI, and C. LASCHI. A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Tokyo, Japan: IEEE, 2013, pp. 5033-5039. DOI: 10.1109/IROS.2013.6697084.

[29] UYGUROGLU, M. and H. DEMIREL. Kinematic analysis of tendon-driven robotic mechanisms using oriented graphs. *Acta mechanica*. 2006, vol. 182, iss. 3-4, pp. 265–277. DOI: 10.1007/s00707-005-0298-z.

[30] DUAN, Y., L. LI, G. JI, and Y. CAI. Vanilla feed-forward neural networks as a discretization of dynamic systems. Available at: `https://arxiv.org/abs/2209.10909`.

[31] DARBY, M. L. and M. NIKOLAOU. Mpc: Current practice and challenges. *Control Engineering Practice*. 2012, vol. 20, iss. 4, pp. 328–342. DOI: 10.1016/j.conengprac.2011.12.004.

[32] SCHWENZER, M., M. AY, T. BERGS, and D. ABEL. Review on model predictive control: An engineering perspective. *The International Journal of Advanced Manufacturing Technology*. 2021, vol. 117, iss. 5-6, pp. 1327–1349. DOI: 10.1007/s00170-021-07682-3.

[33] BADGWELL, T. A. and S. J. QIN. Model Predictive Control in Practice. *Cham: Springer International Publishing*. 2021, pp. 1239–1252. DOI: 10.1007/978-3-030-44184-5_8.

[34] GUPTA, S., A. KUMAR, N. S. TRIPATHY, and S. V. Shah. Rl-based variable horizon model predictive control of multi-robot systems using versatile on-demand collision avoidance. Available at: `https://arxiv.org/abs/2308.07071`.

[35] KHIDHIR, A. H. M. Y. G. Real-time end-to-end self-driving car navigation. *International Journal of Intelligent Systems and Applications in Engineering*. 2023, vol. 11, no. 2s, p. 366–372. . Available at: `https://ijisae.org/index.php/IJISAE/article/view/2732`.