U.P.B. Sci. Bull., Series D, Vol. 84, Iss. 3, 2022 ISSN 1454-2358

NONLINEAR MODEL PREDICTIVE CONTROL FOR TRAJECTORY TRACKING OF A CLASS OF CONTINUUM ROBOTS

Ammar AMOURI, Halim MERABTI, Abdelhakim CHERFIA, and Yazid LAIB DIT LEKSIR

This paper presents a Nonlinear Model Predictive Control (NMPC) scheme for solving the trajectory tracking and obstacle avoidance problems for a class of continuum robots with three actuators per bending section, namely Cable-Driven Continuum Robot (CDCR). Since, NMPC schemes were strongly limited by the computational burden associated with the optimization algorithms, the Knowledge-based Particle Swarm Optimization (KPSO) algorithm is used to solve the existing optimization problem in the NMPC, due to its simplicity and fast convergence. The proposed NMPC-KPSO has been applied to the kinematic models developed on the basis of kinematic equations of inextensible bending section and by using the Constant Curvature Kinematic Approach (CCKA). The proposed control scheme was evaluated via simulation examples with complex trajectories in a free and confined environment. The obtained results showed satisfactory performances in terms of tracking accuracy, computation time and obstacle avoidance. Considering the quality of solution and computation time, the proposed NMPC-KPSO can be considered as an alternative solution for real-time applications of this class of continuum robots.

Keywords: Continuum robot, cable-driven continuum robot, nonlinear model predictive control, particle swarm optimization, trajectory tracking, obstacle avoidance.

1. Introduction

Continuum robots are a category of hyper-redundant manipulators that are inspired by the biological world like elephant trunks [1], snakes [2] and octopus tentacles [3]. These robots are characterized by high dexterity and

[1]Assistant Professor, Department of Mechanical Engineering, Laboratory of Mechanics, Frères Mentouri Constantine 1 University, Algeria, e-mail: ammar_amouri@yahoo.fr

[2]Senior Researcher, Research Center in Industrial Technologies, CRTI, P. O. Box 64, Cheraga 16014, Algiers, Algeria, e-mail: merabtihalim@yahoo.fr

[3]Assistant Professor, Department of Mechanical Engineering, Laboratory of Mechanics, Frères Mentouri Constantine 1 University, Algeria, e-mail: cherfia_abdelhakim@yahoo.fr

[4]Assistant Professor, Department of Mechanical Engineering, University of L'Arbi Ben M'hidi, Oum el Bouaghi, Algeria, e-mail: yaziddl@yahoo.fr

flexibility allowing them to operate in confined spaces and complex environments where conventional rigid robots cannot work. These properties make continuum robots suited for a large number of applications in various fields, such as the medical field, in particular for minimally invasive surgery [4], rescue operation [5], and so on. Nowadays, several designs of the continuum robot have been built [6, 7]. The popular hard class with three actuators per bending section, namely Cable-Driven Continuum Robot (CDCR) [8, 9] is one of them. A CDCR can comprise single or multi-bending sections. Each bending section can be bent and rotated with respect to other sections to control the robot shape and end-tip pose giving two Degrees of Freedom (DoFs).

In literature, classical and advanced control schemes have been widely used to control different robotic systems, such as wheeled mobile robots, serial/parallel robot manipulators, and unmanned aerial vehicles. Nevertheless, with regards to continuum robots, the design of control schemes for these robotic systems can be considered as a challenge due to the complexity of their mathematical models, hyper-redundancy and modeling inaccuracies. However, some works relative to the control of continuum robots have been proposed using different control strategies, such as Proportional-Integrated-Derivative (PID) controller, Adaptive Control Algorithms (ACAs), Fuzzy Logic Controller (FLC) and Nonlinear Model Predictive Control (NMPC).

Regarding classical controllers, two contributions have been made on CDCRs [10, 11], where the authors used a PID controller allowing accurate tracking of trajectories using approximate dynamic models for a single and multi-bending sections continuum robot. However, the robustness of continuum robots control depends on the accuracy of their models; some contributions have been proposed using advanced control schemes. For instance, adaptive Neural Networks (NNs) scheme [12] and adaptive Support Vector Regressor (SVR) controller [13] have been made on a continuum manipulator, namely Compact Bionic Handling Arm (CBHA). In [14], a fuzzy logic based static feedback controller is developed for a single bending section of a Tendon-Driven Continuum Robot (TDCR); while in [15], a FLC is proposed for autonomous execution of end-effector trajectory tracking tasks for a continuum manipulator, namely TDCR, using the kinematic model.

Other researchers have used NMPC to control the continuum robots [16, 17]. In [16], authors propose a NMPC scheme to control the growth of vine-like growing robots in which the growth control was applied to the kinematic model and in [17], where a MPC has been developed for the autonomous steering of concentric tube robots.

In the present paper, we attempt to apply a NMPC-KPSO to control a class of continuum robots, namely Cable-Driven Continuum Robot (CDCR). The proposed control scheme gives a better performance in terms of rapid convergence, and trajectory tracking accuracy via point-to-point technique in a free and confined environment. It is noteworthy that the main advantage of KPSO algorithm is that it allows computing the control signal faster than other optimizer methods [18].

The remainder of this paper is organized as follows: Section 2 gives a brief summary of the forward and differential kinematic models of a CDCR based on CCKA . Section 3 describes the Nonlinear Model Predictive Control and the optimizer method, namely Knowledge-based Particle Swarm Optimization. Simulation results and analysis are presented in Section 4. Concluding remarks and future works are provided in Section 5.

2. Kinematical Models of CDCR

Among the methods proposed to tackle the forward kinematics of continuum robots, the constant curvature kinematic approach (CCKA) [9, 19] is commonly used, due to its simplification in modeling. This section provides a brief summary of the kinematics of a CDCR based on CCKA. Since the CDCR is an open kinematic chain of serially connected bending sections, the position and orientation of the robot's end-tip can be expressed as follows:

$$\mathbf{x} = [\mathbf{x}^{\mathrm{P}}\ \mathbf{x}^{\mathrm{R}}]^T = h(\mathbf{k}), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^6$ is the task variables of the CDCR's end-tip. $\mathbf{x}^{\mathrm{P}} \in \mathbb{R}^3$ labels the Cartesian position vector, and $\mathbf{x}^{\mathrm{R}}$ presents the rotation angles $\{\Phi,\ \Theta,\ \Psi\}$ of the frame attached to the robot's end-tip with respect to the reference frame. $\mathbf{k} = [\mathbf{k}_1^T\ \ldots\ \mathbf{k}_n^T]^T \in \mathbb{R}^{n\times 2}$ is the configuration state of the whole CDCR, and $\mathbf{k}_j = [\theta_j \quad \varphi_j]^T$ is a vector of bending and orientation angles which are used as generalized coordinates (i.e. inputs/outputs) to control the robot. h is a robot-independent mapping function (For more details on the development of forward and differential kinematics of the considered robot, we refer the reader to reference [20, 21]).

The derivative with respect to time of the task variables, $\mathbf{x}$, yields the CDCR's end-tip Cartesian and angular velocity as follows:

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{k})\dot{\mathbf{k}}, \tag{2}$$

where $\mathbf{J}(\mathbf{k}) \in \mathbb{R}^{6\times 2n}$ is the Jacobian matrix that can be calculated as follows:

$$\mathbf{J}(\mathbf{k}) = \frac{\partial h(\mathbf{k})}{\partial \mathbf{k}}, \tag{3}$$

3. Nonlinear model predictive control based on the KPSO

This section gives a summary of the NMPC strategy and the applied optimizer method, namely KPSO. The NMPC-KPSO is applied to the kinematic model i.e. the manipulated variables are the velocities in the configuration space, which is proposed to control the CDCR under study. The proposed NMPC-KPSO scheme will allow for accurate tracking of reference trajectory defined in a task space by considering the imposed physically constraints as well as presence/absence of environmental constraints. Beginning with the mathematical formulation of NMPC, the KPSO and the problem setting are described thereafter.

3.1. Mathematical Formulation of NMPC

For any robotic system, the kinematic motion can be described by the discrete state space model as follows:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)), \tag{4}$$

where $\mathbf{x} \in \mathbf{X}$ is the constrained state space in a convex and closed set. $\mathbf{u} \in \mathbf{U} \subset \mathbb{R}^n$ is the constrained control signal in a compact convex set, and $\mathbf{f}$ is a continuous mapping with $\mathbf{f}(0,0) = 0$.

To regulate the state to the origin by using the NMPC, the optimization problem to be minimized can be expressed as follows:

$$\mathbf{C}_N(\mathbf{x}, k, \mathbf{u}) = \omega(\mathbf{x}_{k+N}) + \sum_{\xi=k}^{k+N-1} f_N(\mathbf{x}_\xi, \mathbf{u}_\xi), \tag{5}$$

where N is the prediction horizon, and $\omega(\mathbf{x}_{k+N})$ is a weight on the final state space. It should be noted that the weight ω and the final region are introduced to guarantee stability of the NMPC.

The solution of the optimization problem described in Equation (5) allows an accurate tracking of a reference trajectory and permits to calculate the control signal across the prediction horizon. At each sampling instant, the problem is solved to obtain the optimal control sequences $\mathbf{u} = [\mathbf{u}_k \; \mathbf{u}_{k+1} \; \ldots \; \mathbf{u}_{k+N-1}] \in \mathbf{U}^N$, but only the first sequence is applied to the robot.

3.2. KPSO

The PSO developed by Kennedy and Eberhart is a metaheuristic optimization method that is inspired by the behavior of particles within a group [22]. The main advantage of this method is its fast convergence with few tuned parameters compared with various global optimization algorithms [18, 23, 24].

In each iteration t, the position of each particle p is updated as a function of the local best position P^t_{pbest} and the global best position P^t_{gbest}, meanwhile each step, the particles positions and velocities are updated according to the following equations:

$$v_p^{t+1} = v_p^t + c_1\rho_1(P^t_{pbest} - x_p^t) + c_2\rho_2(P^t_{gbest} - x_p^t), \tag{6}$$

$$x_p^{t+1} = x_p^t + v_p^{t+1}, \tag{7}$$

where x_p^t and v_p^t are the position and velocity of the particle p; c_1 and c_2 are constants; ρ_1 and ρ_2 are random numbers between 0 and 1.

Since a common barrier to optimization methods is the early convergence of solution to local minima that makes them inefficient in practical applications; to tackle this issue, the re-generating technique of initial population of the swarm [25] has been added to the KPSO method in this paper, see Fig. 1.

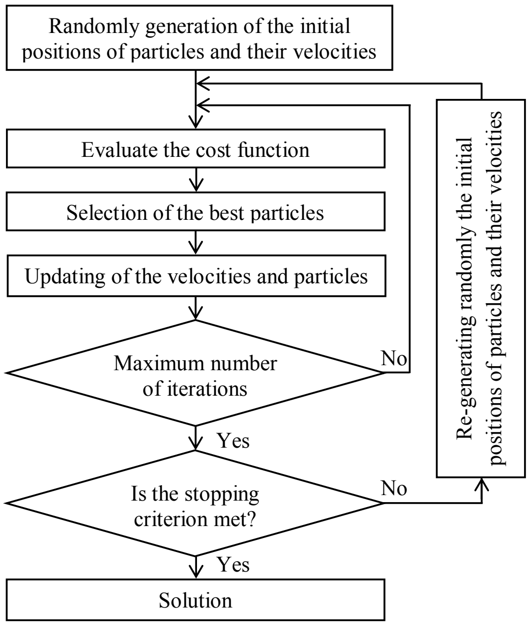


Fig. 1 Block Diagram of Applied KPSO method

3.3. Problem setting

The purpose of this study is to find the control signals, i.e. bending and orientation velocities of each bending section, by using the NMPC-KPSO that allows the robot to track a given reference trajectory in its workspace. Since the optimization problem arising in the NMPC is generally non convex, the use of metaheuristic methods may be justified. A prior knowledge of the best solutions allows the metaheuristic method to reduce the computation time and increase the solution precision [18, 23, 24]. Thus, in this paper, the Knowledge-based Particle Swarm Optimization (KPSO) method is used to minimize the cost function defined in equation (5) as in works [26, 27]. The coordinates of the particle in PSO are the variables of the optimization problem, which represent in our case the angular velocity of the bending and orientation angles of each bending section. In three-dimensional search space,

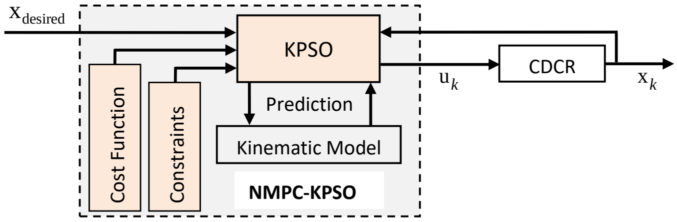


Fig. 2 Block diagram of the proposed NMPC-KPSO

the dimension of the optimization problem is $(2 \times n)$. The block diagram of the proposed NMPC-KPSO scheme is shown in Fig. 2.

When obstacles exist in the robot's workspace, the strategy can be carried out by placing some points on the robot's central axis so that these points do not collide with existing obstacles and the robot reaches the target. To realize obstacle avoidance, a given penalty or some constraints conditions will be added to the cost function. The obstacles are identified by a position vector with respect to reference frame $\mathcal{R}_0$. To Sum up, the problem is to find the control law defined by $\dot{\theta}_j$ and $\dot{\varphi}_j$, with $j = 1, 2, \ldots, n$, that allows the robot to: (1) track a given reference trajectory defined in the task space, and (2) avoid the static obstacles existing in its workspace.

4. **Simulations and analysis**

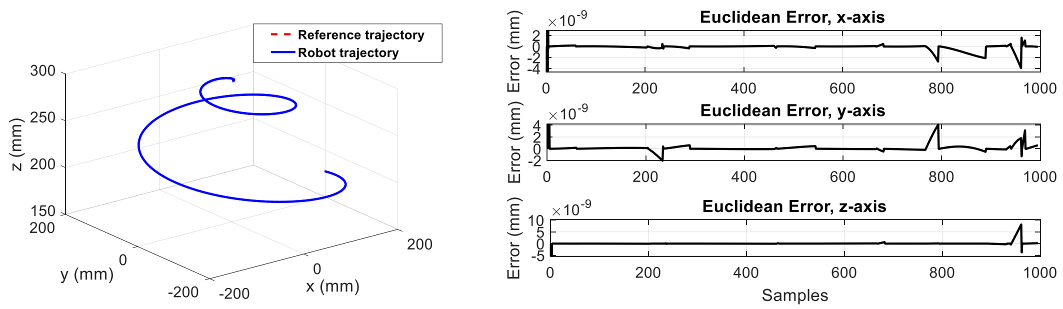
In order to test the feasibility and verify the performance of the proposed NMPC-KPSO scheme in terms of tracking accuracy and computation time, three simulation examples for tracking spatial trajectories, in a free and confined environment and in the presence of static obstacles, are performed. The first example is considered to implement the proposed controller on a one-, two- and three-bending section CDCR for tracking a spiral-shaped trajectory, inside their workspaces, in a free environment i.e. without obstacles. The second one represents a robot with three-bending section tracking a line-shaped trajectory with a specific orientation for the end-tip of the CDCR. The third example concerns a CDCR with one-bending section for tracking a circular-shaped trajectory in the presence of a static obstacle located on the reference trajectory. The lengths of all CDCR sections are assumed physically identical and share the values for ℓ_j, with $j = 1, 2, 3$, as 300 mm. The selected parameters of the KPSO method which offer an acceptable compromise on performance are: swarm size = 6, iterations = 30 and $c_1 = c_2 = 1.2$. The simulation examples are conducted on MATLAB software using Intel® Core^{TM} i3-2310M CPU at 2.10 GHz and 4GB RAM.

4.1. **Simulation 1: trajectory tracking in a free Cartesian space**

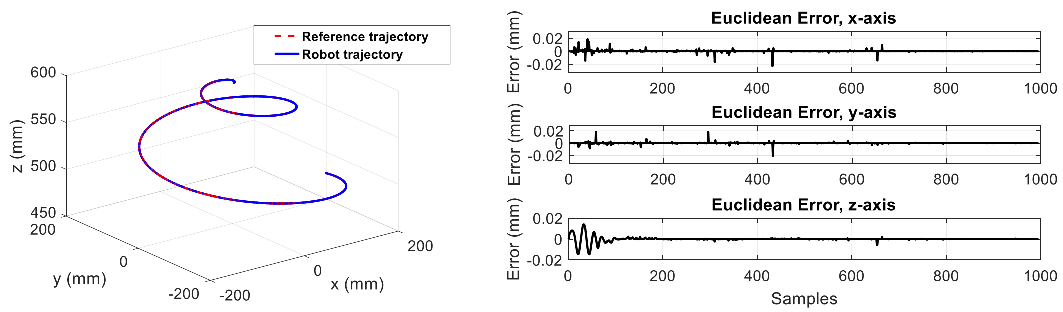
In this simulation, the spiral-shaped trajectories defined by Equation (8) are considered to evaluate the performances of the proposed NMPC-KPSO scheme in terms of trajectory accuracy and computation time. To achieve this purpose, the same form of reference trajectory is used as input for CDCRs consisting of one-, two- and three-bending section. All robots start from an initial state as $\mathbf{x}^{\mathrm{P}} = [1.7387,\ 01521,\ \mathrm{P}_z + 299.9932]^T$ where P_z is equal to 0 mm, 300 mm and 600 mm, respectively.

$$\mathbf{x}^{\mathrm{P}} = \left\{ h(t).\sin\left(\frac{t}{15}\right).\cos(t),\ h(t).\sin\left(\frac{t}{15}\right).\sin(t),\ \mathrm{P}_z + h(t).\cos\left(\frac{t}{15}\right) \right\}^T, \quad (8)$$

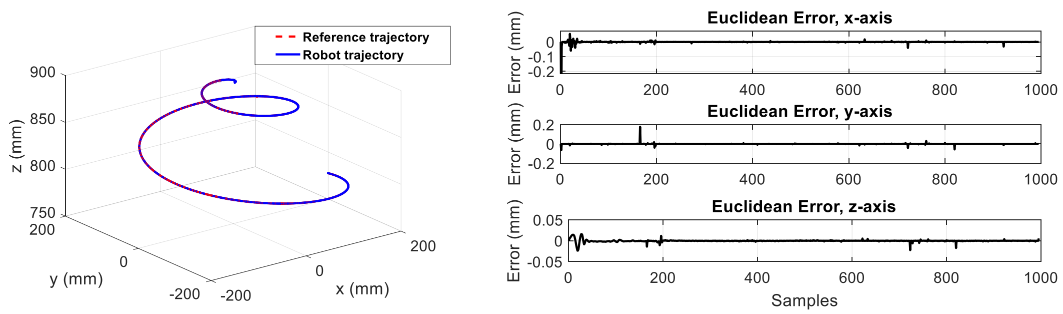
where $h(t) = 15\ell_j.\sin(t/15)/t$. The controller time step is chosen to be $t = 0.005$ (sec) and the prediction horizon is $N = 2$ with a total simulation samples is equal to 994.



(a) One-bending section



(b) Two-bending section



(c) Three-bending section

Fig. 3 Actual and reference trajectories, and Euclidean errors between them

The simulation results in Figs. 3(a), 3(b) and 3(c) show the tracking of the desired spiral-shaped trajectory for one-, two- and three-bending section CDCR, respectively, and Euclidean errors between the actual and reference trajectory along x-axis, y-axis and z-axis. From these Figures, it can be seen that the curves are almost superposed and that the average errors for three cases are respectively smaller than 5.10^{-9} mm, 7.10^{-4} mm and 5.10^{-3} mm, see

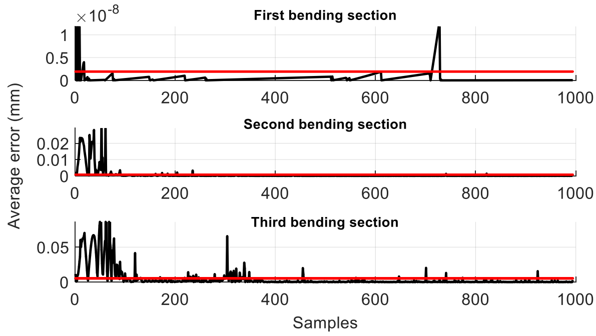


Fig. 4 Average trajectories tracking errors

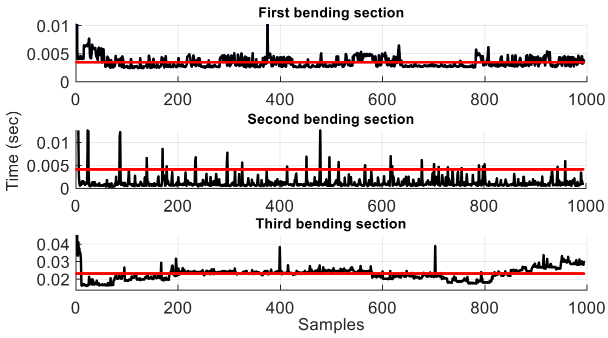


Fig. 5 Computation times for tracking the spiral-shaped trajectories

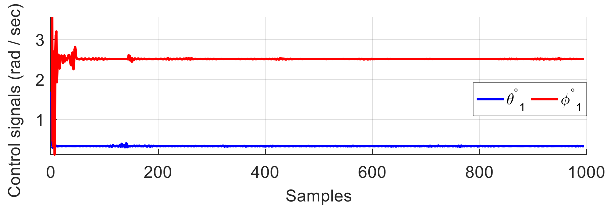
Fig. 4. The computation times for three cases are presented in Fig. 5 where the mean values are smaller than 4 msec, 5 msec and 23 msec, respectively. These results demonstrate the good tracking of the reference trajectories with very low computation times. The required control signals to track the spiral-shaped trajectories, for three cases, are shown in Fig. 6.

4.2. Simulation 2: trajectory tracking with a specified orientation

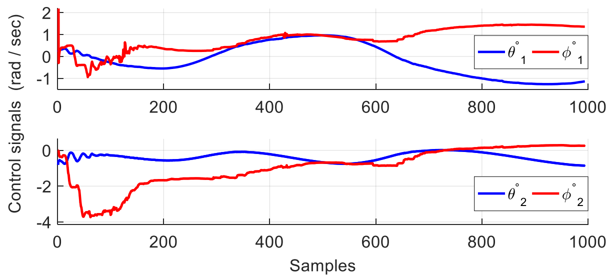
For most applications of such a robotic system, both positioning and orientation of the end-effector are important. This simulation presents results for tracking a trajectory with a specific orientation assigned to the end-tip of CDCR. For this, the CDCR equipped with three-bending section is used for tracking a line-shaped trajectory defined by Equation (9) with end-tip' orientations of $\{\Phi = 0,\ \Theta = 0,\ \Psi = 0\}$.

$$\mathbf{x}^{\mathrm{P}} = \{30t,\ 0,\ 900 - 10t\}^T, \tag{9}$$

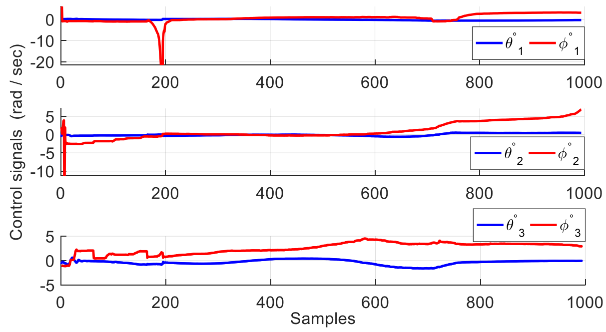
Fig. 7 presents some configurations of the CDCR for the line-shaped trajectory tracking with previously-defined orientations. The simulation was carried out by adding additional constraints defined as $|\Theta, \Psi| \leq 10^{-2}$ (rad) to



(a) One-bending section



(b) Two-bending section



(c) Three-bending section

Fig. 6 Control signals for tracking the spiral-shaped trajectories

the cost function. We note, here, that these CDCR' configurations are achieved using the kinematic model in which the required bending and orientation angles (see Fig. 8) are obtained by analytical integration of the control signals shown in Fig. 9. The Euclidean errors between the actual and reference trajectory, along x-axis, y-axis and z-axis, as well as errors of end-tip' orientations are shown in Fig. 10. From this Figure, it can be seen that there is a significant convergence between the actual and reference trajectory where the average error is less than 0.016 mm. The computation times is depicted in Fig. 11 where the mean value is less than 68 msec. In this simulation the robot starts from an initial state of $t = 0$ with a total simulation sample of 510, and the parameters of the applied KPSO method are chosen as: iterations $= 70$ and swarm size $= 15$.

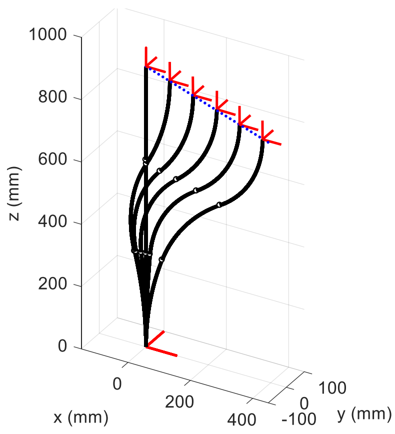


Fig. 7 Some configurations of CDCR showing the tracking a line-shaped trajectory with an orientation of $\{\Phi = 0,\ \Theta = 0,\ \Psi = 0\}$

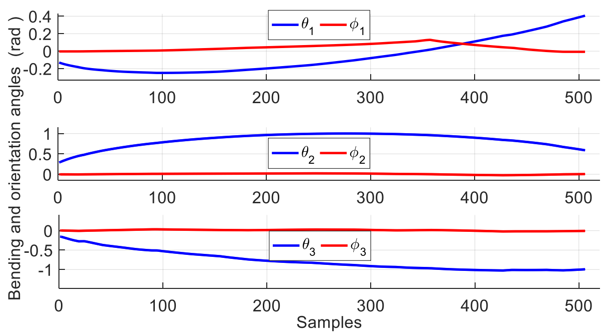


Fig. 8 Required bending and orientation angles to track the line-shaped trajectory with specified orientations of the CDCR's end-tip

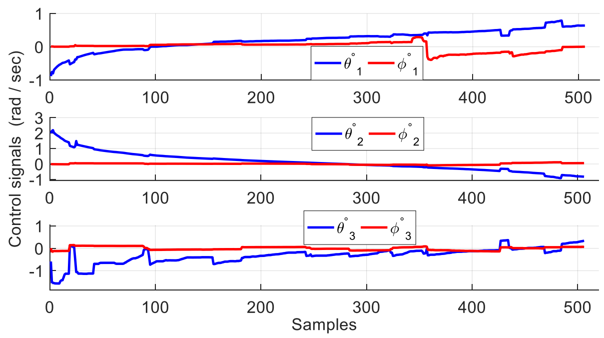


Fig. 9 Control signals to track the line-shaped trajectory with a specific orientation of the CDCR's end-tip

4.3. Simulation 3: trajectory tracking in Cartesian space in the presence of obstacle

In this simulation, the CDCR composed of one-bending section tracks a circular-shaped trajectory defined as $\mathbf{x}^{\mathrm{P}} = [169.86\cos(t),\ 169.86\sin(t),\ 221.37]^T$

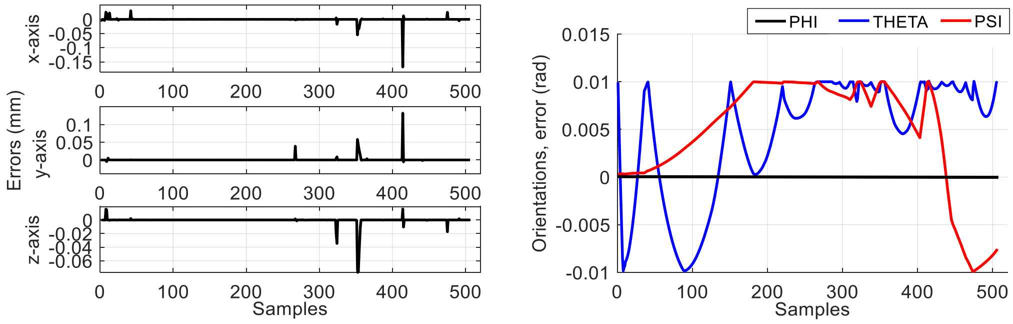


Fig. 10 Euclidean errors between the actual and reference line-shaped trajectory and end-tip's orientations errors

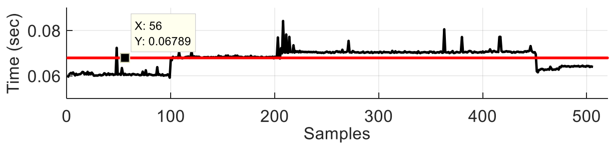


Fig. 11 Computation time for tracking the line-shaped trajectory with a specified orientation

(mm) in the presence of a static obstacle which is located on the reference trajectory at $\mathbf{x}^{\mathrm{P}}(t = \pi/2)$. The robot starts from an initial state as $\mathbf{x}^{\mathrm{P}} = [10^{-5},\ 10^{-5},\ 300]^T$ (mm). The simulation results for tracking the above trajectory while avoiding the fixed obstacle are shown in Fig. 12. From this figure, it can be seen the good tracking of the reference trajectory by the robot and the success to avoid a static obstacle. The obstacle is avoided by adding a penalty to the cost function where the distance between the robot and the static obstacle is less than a given safe distance. The necessary variation of the cables length to track the desired trajectory and avoid the static obstacle is depicted in Fig. 13. In this simulation, the mean computation time is less than 2.6 msec (see Fig. 14) which is very encouraging for real time applications.

As a general conclusion to our analysis, first, the results presented in Figs. 3, 4, 10 and 12 point out a good accuracy and a high quality of the trajectory tracking of the NMPC-KPSO scheme via point-to-point technique. Second, Fig. 5, 11 and 14 show the low computation times associated to the trajectory tracking with/without obstacle avoidance and with a specified orientation of the robot's end-tip. On the other hand, the computed control signals are very smooth (see Figs. 6, 9 and 13). Based on these obtained results, the proposed NMPC-KPSO scheme can be considered as an alternative solution for real-time applications for this class of continuum robots.

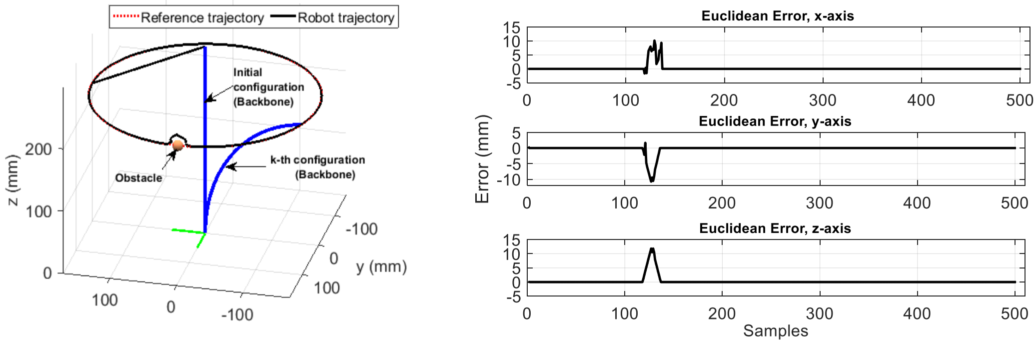


Fig. 12 Actual and reference trajectory and Euclidean errors between them for a circular-shaped trajectory tracking in the presence of a static obstacle

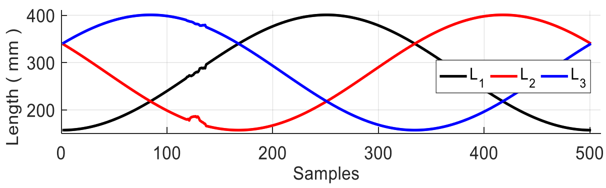


Fig. 13 Profile of the cables length for tracking the circular-shaped trajectory in the presence of a static obstacle

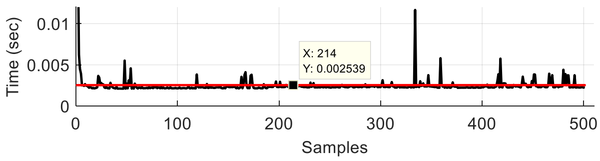


Fig. 14 Computation time for tracking the circular-shaped trajectory in the presence of a static obstacle

5. Conclusion

In this paper, a Nonlinear Model Predictive Control (NMPC) scheme was used to control a class of continuum robots, namely Cable-Driven Continuum Robot (CDCR). The Knowledge-based Particle Swarm Optimization (KPSO) method was used for the solution of optimization problem arising in NMPC. The NMPC-KPSO scheme has been applied to kinematic models developed on the basis of kinematic equations of inextensible bending section by using the Constant Curvature Kinematic Approach (CCKA). The proposed NMPC-KPSO scheme is simulated over different simulation examples ranging from the trajectory tracking in free Cartesian space with/without specific orientation and with the presence of obstacle. The quality of solution, in terms of trajectory tracking and smoothness of the control signals as well as the computation times show that the proposed NMPC-KPSO scheme is a feasible alternative for real-time applications. As a perspective of this work, we intend to apply the proposed scheme for dynamic obstacles as well as a possible extension to a case where dynamic models are used instead of kinematic models in three-dimensional space.

Acknowledgments

The authors thank the anonymous referees for their careful reading of our manuscript and their many insightful comments and suggestions, which allowed to correct and improve the paper.

References

[1] *M. W. Hannan and I. D. Walker*, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," Journal of Robotic Systems, **vol. 20**, 2003, pp. 13-20.

[2] *G. S. Hirose and M. Mori*, "Biologically inspired snake-like robots," in Proceedings of IEEE International Conference on Robotics and Biomimetics, Shenyang, China, 2004, pp. 1–7.

[3] *I. S. Godage, E. Guglielmino, D. T. Branson et al.*, "Novel modal approach for kinematics of multisection continuum arms," in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 2011, pp. 1093–1098.

[4] *J. Burgner-Kahrs, D. C. Rucker and H. Choset*, "Continuum robots for medical applications: A survey," IEEE Transactions on Robotics, **vol. 31**, no. 6, 2015, pp. 1261–1280.

[5] *H. Tsukagoshi, A. Kitagawa and M. Segawa*, "Active hose: an artificial Elephant's nose with maneuverability for rescue operation," in Proceedings of IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001, pp. 2454–2459.

[6] *S. Kolachalama and S. Lakshmanan*, "Continuum robots for manipulation applications: A Survey," Journal of Robotics Volume 2020, Article ID 4187048, 19 pages (`https://doi.org/10.1155/2020/4187048`).

[7] *Y. Zhang and M. Lu*, "A review of recent advancements in soft and flexible robots for medical applications," International Journal of Medical Robotics and Computer Assisted Surgery, **vol. 16**, no. 3, 2020, https://doi.org/10.1002/rcs.2096

[8] *A. Amouri*, "Investigation of the constant curvature kinematic assumption of a 2-Dofs cable-driven continuum robot," UPB Scientific Bulletin, Series D: Mechanical Engineering, **vol. 81**, no. 3, 2019, pp. 27–38.

[9] *I. A. Gravagne and I. D. Walker*, "Kinematic transformations for remotely-actuated planar continuum robots," in Proceedings of IEEE International Conference on Robotics and Automation, San Francisco, CA, 2000, pp. 19–26.

[10] *A. Amouri, A. Zaatri, and C. Mahfoudi*, "Dynamic modeling of a class of continuum manipulators in fixed orientation," Journal of Intelligent and Robotic Systems, **vol. 91**, no. 3-4, 2018, pp. 413–424.

[11] *A. Amouri, C. Mahfoudi, and A. Zaatri*, "Dynamic modeling of a spatial cable-driven continuum robot using Euler-Lagrange method," International Journal of Engineering and Technology Innovation, **vol. 10**, no. 1, 2020, pp. 60–74.

[12] *A. Melingui, O. Lakhal, B. Daachi et al.*, "Adaptive neural network control of a compact bionic handling arm," IEEE/ASME Transactions on Mechatronics, **vol. 20**, no. 6, 2015, pp. 2862–2875.

[13] *A. Melingui, J. J. Mvogo Ahanda, O. Lakhal et al.*, "Adaptive Algorithms for Performance Improvement of a Class of Continuum Manipulators," IEEE Transactions on Systems, Man, and Cybernetics: Systems, **vol. 48**, no. 9, 2018, pp. 1531–1541.

[14] *W. Ba, X. Dong, A. Mohammad et al.*, "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, 2021, pp. 3010–3021.

[15] *P. Qi, C. Liu, A. Ataka et al.*, "Kinematic control of continuum manipulators using a fuzzy-model-based approach," IEEE Transactions on Industrial Electronics, **vol. 83**, no. 8, 2016, pp. 5022–5035.

[16] *H. El-Hussieny, I. A. Hameed and J. H. Ryu*, "Nonlinear model predictive growth control of a class of plant-inspired soft growing robots," IEEE ACCESS, **vol. 8**, 2020, pp. 214495–214503.

[17] *M. Khadem, J. O'Neill, Z. Mitros et al.*, "Autonomous steering of concentric tube robots via nonlinear model predictive control," IEEE Transactions on Robotics, **vol. 36**, no. 5, 2020, pp. 1595–1602.

[18] *J. Zietkiewicz, P. Kozierski and W. Giernacki*, "Particle swarm optimization in nonlinear model predictive control; comprehensive simulation study for two selected problems," International Journal of Control, **vol. 94**, no. 10, 2020, pp. 2623–2639.

[19] *R. J. Webster and B. A. Jones*, "Design and kinematic modeling of constant curvature continuum robots: A review," International Journal of Robotics Research, **vol. 29**, no. 13, 2010, pp. 1661–1683.

[20] *A. Amouri*, "Contribution à la modélisation dynamique d'un robot flexible bionique," Ph.D. dissertation, University of Constantine 1, Algeria, July 2017. http://depot.umc.edu.dz/handle/123456789/6489

[21] *C. Escande*, "Towards modeling of a class of bionic manipulator robots," Ph.D. dissertation, University of Sciences and Technologies of Lille, France, 2013. http://www.theses.fr/2013LIL10171

[22] *J. Kennedy and R. C. Eberhart*, "Particle swarm optimization," in Proceedings of IEEE Iinternational Conference on Neural Networks IV, 1995, pp. 1942–1948.

[23] *A. Amouri, C. Mahfoudi, A. Zaatri, et al.*, "A metaheuristic approach to solve inverse kinematics of continuum manipulators," Journal of Systems and Control Engineering, **vol. 231**, no. 5, 2017, pp. 380–394.

[24] *A. R. Jordehi and J. Jasni*, "Parameter selection in particle swarm optimization: A survey," Journal of Experimental and Theoretical Artificial Intelligence, **vol. 45**, no. 4, 2013, pp. 527–542.

[25] *R. V. Ram, P. M. Pathak and S. J. Junco*, "Inverse kinematics of mobile manipulator using bidirectional particle swarm optimization by manipulator decoupling," Mechanism and Machine Theory, **vol. 131**, 2019, pp. 385–405.

[26] *H. Merabti and K. Belarbi*, "Accelerated micro particle swarm optimization for the solution of nonlinear model predictive control," World Journal of Engineering, **vol. 14**, 2017, pp. 509-521.

[27] *H. Merabti, B. Bouchemal, K. Belarbi, et al.*, "Non linear predictive control of a LEGO mobile robot," AIP Conference Proceedings, **vol. 1618**, no. 1, 2014, pp. 647-650.