A Serpentine Curve Based Motion Planning Method for Cable Driven Snake Robots

Lei Tang, Li-Min Zhu, Xiangyang Zhu and Guoying Gu

***Abstract*—Cable driven snake robots (CDSRs) have the ability to slither around the narrow space owing to high degrees of freedom (DOFs). Due to the high DOFs, the inverse kinematics of the snake robot is nonlinear and has no unique solution, which makes the motion planning quite difficult. In this paper, a serpentine curve based method is presented to simplify the motion planning process. Firstly, the configuration of the CDSR is represented by a serpentine curve. Secondly, the curvature of the serpentine curve is determined to satisfy the obstacle constraints. Finally, the CDSR moves along the planned curves at each control period. The serpentine curve has much less parameters than the DOFs of the robot. Letting the robot slither along the planned curves, the dimensions of motion planning can be reduced significantly. Simulation and experiment on a 25 DOFs CDSR prototype show that the presented method works well on the designed CDSR platform.**

I. Introduction

CDSRs are of increasingly interest for their potential capability in applications in constraint environments. Researches have been done in such kinds of situations, for example inspection and maintenance of complex industrial devices such as aircraft wings [1], engines [2], nuclear reactors and pipelines [3], [4], and minimally invasive surgery [5]–[7].

In order to conduct operations in the above mentioned confined environments, the CDSRs often occupy slender structure and high DOFs [8], which generates high dexterity and maneuverability to deal with abundant obstacles in constraint situations. However, with the increasing DOFs of the robot, the motion planning problem of it becomes more and more prohibitive. Because the inverse kinematics and control of snake robots become increasingly complicated with each added DOF [9]–[12]. Especially when the robot is kinematic hyper-redundant which means the number of DOFs exceeds the dimension of the task space greatly.

The inverse kinematics is indispensable for a robot to conduct motion planning and control. While the high DOFs make the inverse kinematics of the snake robot complex and nonlinear. There are much more unknown variables than the constraint equations. Thus, the solution of the inverse kinematics is not unique, which makes the problem more serious. Traditionally, kinematic analysis and motion planning for snake robots has relied upon a pseudoinverse [13], [14], or extended inverse [15] of the Jacobian matrix. These schemes, however, are difficult to the real time control of hyper-redundant snake robots, because of their computational inefficiency. To control the hyper-redundant snake robots in real time, curvilinear based theories are exploited by [16]–[18] and Hirose [19]–[21]. They utilize curves to describe the configuration of snake robots. The curves are parametrized by some coefficients and curvature functions. Chirikjian named the curvature functions modes, and the method is called modal approach. Hirose and Ma [17], [20] also proposed the similar method. They employed continuous curves to model the posture of snake robots with curvature functions, and named the method position-coordination scheme or serpentine curve method. The continuous curves that represent the configurations of snake robot have much less parameters than the DOFs of the robot. Therefore, the dimensions of the motion planning reduce significantly by planning the curves and letting the robot slither along them. Thus, the inverse kinematics and motion planning problem of the snake robot can be solved efficiently.

Many snake robots are discretely actuated, which means they can only move at the joint. However, it is vital to let the robot match the planned serpentine curve well. Various algorithms are put forward to complete this task. A least-square matching method is presented in [22], in which a fitting error function is composed with some constraints and the fitting is achieved by minimization of the composed function. Another method based on recursive backward solution of single nonlinear equations is presented in [23]. This method matches the position of the last joint with the target point and find the other position of the joints one by one in a backward way with matching the joint approximate to the continuous curve according to the link length restrictions. Andersson [24] derived explicit expressions for joint angles in a product of exponential formula way also by searching along the planned curves.

In summary, the motion planning for CDSR is difficult due to its high DOFs. The Jacobian based algorithms are of low computational efficiency. Therefore, this work is motivated to develop a serpentine curve based motion planning method for CDSR. Firstly, the configuration of CDSR is represent by a serpentine curve. Secondly, the curvature functions are defined to negotiate the obstacle constraints. The serpentine curve is determined according to the tunnel curvature function. Finally, the CDSR is fitted to the planned curve and move along it. Although some curve based methods have been developed for snake robots, there are rarely experiment validation. The method presented in this paper is verified on a 25 DOFs CDSR prototype.

The rest of this paper is organized as follows: Section II describes the overall system and kinematics of the designed CDSR. Section III presents the serpentine representation of the CDSR. Curvature function which satisfies to the obstacle constraints and the curve fitting algorithm are also described in this section. Simulation and case study of presented method are presented in Section V. Section VI shows the experiment results on a 25 DOFs CDSR platform. Section VII concludes this paper.

*This work is supported by the National Natural Science Foundation of China No. 51622506 and 51435010.

The authors are with State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (Corresponding to Guoying Gu guguoying@sjtu.edu.cn).

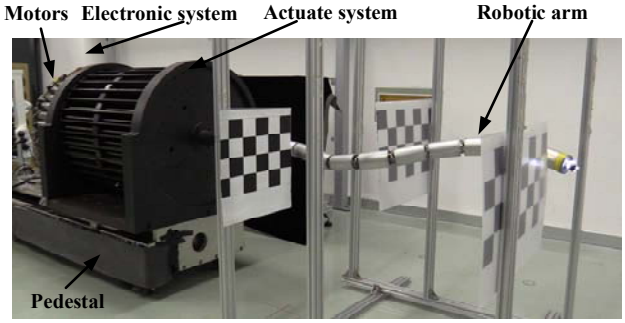

Fig. 1. Schematic of the whole system of the robot.

II. DEVELOPMENT OF CABLE DRIVEN SNAKE ROBOT

The structure overview and kinematics of the developed CDSR are briefly introduced in this section. The reader may refer to [13] for more detailed information.

A. Overview of the cable-driven snake robot

The CDSR consists of a robotic arm, actuate system, electronic system and a linear slide base which are shown in Fig. 1. The robotic arm has hybrid structure, which is composed of 12 parallel mechanism sections connected in a serial way. Each section is a four chains parallel mechanism consisting of three spherical-prismatic-spherical (SPS) chains and a universal joint chain (3-SPS-U). S, P and U represent spherical, prismatic and universe joint, respectively.

The length of each section is 91 mm, which is composed of a 70 mm rigid bar, two 4 mm disks and two 6.5mm joint bars, which are shown in Fig. 2. Each joint consist of a 2-DOF universal joint. The 36 cables are attached to the rear of the robot. All the cables are evenly circumferential distributed. Therefore, the nominal interval between two adjacent cables is 10 degrees. Each section is driven by three cables, which have a nominal interval of 120 degrees. All the cables are routing proximal sections and fasten to the distal disk of distal section. The CDSR has 25 DOFS with 24 rotational DOFs and 1 translation DOF. The maximum joint limit is 27degrees.

B. Kinematics model of cable-driven snake robot

The cable driven snake robot consists of actuate cables, joints and end effector. The lengths of cables are notified as the actuator space. The rotation angles of each section are described as the joint space, and the position and orientation of the tip of the robot are defined as the task space. Since the model from joint space to task space is needed for the development of serpentine curve based motion planning method, we only present the model from joint space to task space. More detailed information can be referred to [13].

As for the serial connected snake arm, each section has two rotational DOFs, and the axis of the two rotational joint are perpendicular to each other. Coordinate frames are establish in each section as shown in Fig. 3.The origin of each joint frame is coincide with the intersection of the two perpendicular axis.

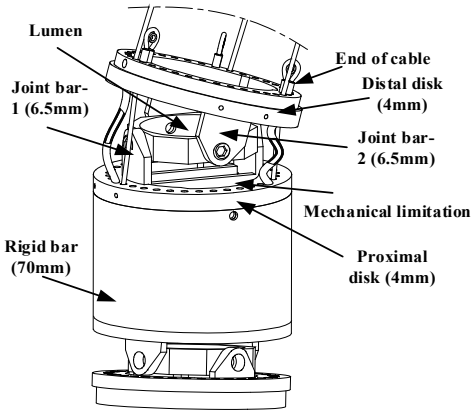

Fig. 2. Schematic of the whole system of the robot.

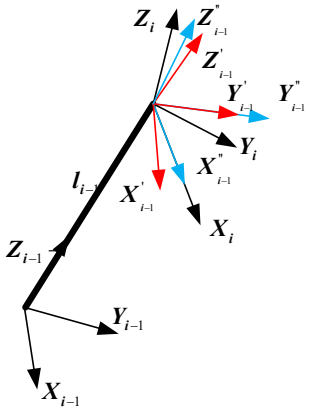

Fig. 3. Schematic of the whole system of the robot

Axes Y_i and X_i coincide with the rotational axes of the first and second joints respectively. Axes Z_i and Z_i are parallel to the central axis of the link in the next section. The axes X_1 and Y_2 are determined according to the right hand rule. Translate frame $X_{i-1}Y_{i-1}Z_{i-1}$ along Z_{i-1} by l_{i-1} to get the frame $X_{i-1}'Y_{i-1}'Z_{i-1}'$. Rotate the new frame about Y_{i-1}' α_i to get the frame $X_{i-1}''Y_{i-1}''Z_{i-1}''$. At last the frame $X_iY_iZ_i$ at the next joint is acquired by rotate the $X_{i-1}''Y_{i-1}''Z_{i-1}''$ frame about X_{i-1}'' β_i.

The transformation between the frames of adjacent joints can be determined as (1).

$$
\begin{aligned}
{}^{i}_{i-1}T &= Trans(0,0,L_{i-1})\,\mathrm{Rot}(y,\alpha_{i-1})\,\mathrm{Rot}(x,\beta_{i-1}) \\
&= \begin{bmatrix}
\cos\alpha_{i-1} & \sin\alpha_{i-1}\sin\beta_{i-1} & \sin\alpha_i\cos\beta_{i-1} & 0 \\
0 & \cos\beta_{i-1} & -\sin\beta_{i-1} & 0 \\
-\sin\alpha_{i-1} & \cos\alpha_{i-1}\sin\beta_{i-1} & \cos\alpha_i\cos\beta_{i-1} & L_{i-1} \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{aligned} \tag{1}
$$

The position and orientation of the end effector can be gotten by the transformation as (2). ${}^{1}_{w}T$ is the transformation matrix of the first link frame related to the world frame.

$$
{}^{n}_{w}T = {}^{1}_{w}T\,{}^{2}_{1}T\cdots{}^{i}_{i-1}T\cdots{}^{n}_{n-1}T \tag{2}
$$

III. SERPENTINE CURVE BASED MOTION PLANNING METHOD

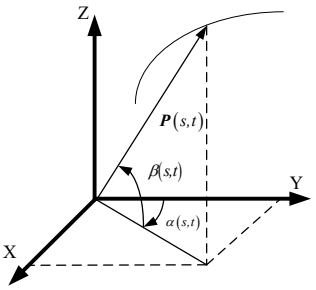

Fig. 4. Parameterization of a point on a spatial curve

A. Serpentine backbone representative of the robot

The CDSRs move dexterously, and have the morphology of some biological objects such as tentacles, elephant trunks and snakes. The configuration of them can be denoted by a series of time varying spatial curve. Usually, the vector of the points on these curves can be determined by some parameters. For example, a spatial curve can be parametrize as in Fig. 4. The mathematic model of the curve can be represented by (3).

$$\begin{aligned} P(s,t) &= [x(s,t), y(s,t), z(s,t)]^T \\ x(s,t) &= \int_0^s \sin(\gamma(s,t))\cos(\varphi(s,t))\,\mathrm{ds} \\ y(s,t) &= \int_0^s \cos(\gamma(s,t))\cos(\varphi(s,t))\,\mathrm{ds} \\ z(s,t) &= \int_0^s \sin(\varphi(s,t))\,\mathrm{ds} \end{aligned} \tag{3}$$

Where $P(s, t)$ is the position vector of a point in the curve and it is determined by $\gamma(s, t)$ and $\varphi(s, t)$, in which s and t are arc length and time, respectively. The frame at the point is shown in Fig. 4. $\gamma(s, t)$ and $\varphi(s, t)$ can be parameterized by some base function and coefficients. The serpentine curves parameterized by trigonometric function own continuous curvature and configuration of biological object. Thus, they are employed to represent the configuration of CDSRs and some other bioinspired robots. Time dependent coefficients a_1, a_2, a_3, a_4 and arc length dependent trigonometric functions are employed to parameterize $\gamma(s, t)$ and $\varphi(s, t)$ which can be denoted in (4).

$$\begin{aligned} \gamma(s,t) &= a_1(t)\sin(2\pi s) + a_2(t)\sin(2\pi s) \\ \varphi(s,t) &= a_3(t)\sin(2\pi s) + a_4(t)\sin(2\pi s) \end{aligned} \tag{4}$$

Given the end position of the curve, the coefficients a_1, a_2, a_3, a_4 can be determined in a numerical way. However, in the case of spatial curve, there is no analytic solution for such coefficients. Therefore, we focus on the plane case. In the plane case, the serpentine curves can be determined by its curvature functions $\kappa(s, t)$. The point on the serpentine curve $P(s, t)$ can be represent by θ (the angle between the tangent vector at this point and the X axis), x and y coordinates, which are denoted in (5).

$$\begin{aligned} P(s,t) &= [\theta(s,t), y(s,t), z(s,t)]^T \\ \theta(s,t) &= \int_0^s \kappa(s,t)\,\mathrm{ds} \\ x(s,t) &= \int_0^s \cos(\theta(s,t))\,\mathrm{ds} \\ y(s,t) &= \int_0^s \sin(\theta(s,t))\,\mathrm{ds} \end{aligned} \tag{5}$$

The curvature functions $\kappa(s, t)$ is also parameterized by time dependent coefficients and arc length dependent trigonometric functions.

B. Serpentine curvature function with obstacle constraints

This algorithm takes the time independent obstacle into consideration. The obstacles in the workspace of the robot is predefined. Therefore, the obstacle avoidance under such circumstance is to find a path around the obstacles. In this sense, the curvature of the path around the obstacles can be predefined. The CDSR move along the predefined path to guarantee the avoidance of the obstacles.

If the base of the robot cannot move, the part of the robot outside the obstacles usually matches to time dependent curves. At the same time, the part inside the obstacles fits to a time independent curve. Thus, the curvature function of the serpentine curve is defined as the free section and constrained section. To achieve this target, window function is used to obtain the piecewise curvature function, which is denoted by (6).

$$\kappa(s,t) = \kappa_f(s,t)W(s,s_s,s_e) + \kappa_c(s,t)W(s,s_s,s_e) \tag{6}$$

Where W is the window imposed on curvature function, s_f and s_e are the start and end points of the free and constrained section respectively. W is defined by (7), where H is the Heaviside step function.

$$W(s,s_s,s_e) = H(s-s_e) - H(s-s_s) \tag{7}$$

The curvature function of the curve in unconstraint environment is denoted by (8). $s_f(t)$ is the arc length of the curve in unconstraint environment.

$$\kappa_f(s,t) = a(t)\cos(2\pi s / s_f(t)) \tag{8}$$

Substitute (8) into (5), the coordinates of the point on the unconstraint curve is (9).

$$\begin{aligned} x(s_f(t),t) &= \int_0^{s_f(t)} \sin\left(\frac{a(t)s_f(t)}{2\pi}\sin\frac{2\pi s}{s_f(t)}\right)ds \\ y(s_f(t),t) &= \int_0^{s_f(t)} \cos\left(\frac{a(t)s_f(t)}{2\pi}\sin\frac{2\pi s}{s_f(t)}\right)ds \end{aligned} \tag{9}$$

C. Curve fitting algorithm

The planned serpentine curve in before sections is continuous. However, for discrete actuated CDSR, the robot should be adhere to the planned curve. In this sense, the center points of each section should be fitted to the planned curve. The center point of the first section is coincide with the begin point of the curve. The other center points P_i of remaining sections should be found along the curve. This problem can be formulate as the following optimization problem.

$$\begin{cases} \min & \left| \left\| P_{i+1} - P_i \right\| - l_i \right| \\ s.t. & P_i \in P(s,t) \end{cases} \tag{10}$$

Where, l_i is the link length of the ith section of the CDSR. Such problem can be solved by the dichotomy method. The angles of joint can be determined according to the positions of the all joint locations (P_i). Z_i is the unit vector of the Z direction of the ith section that can be denoted by (11). The relationship of Z axis of section i+1 and i can be presented by (12).

$$Z_i = \frac{P_{i+1} - P_i}{\left\| P_{i+1} - P_i \right\|} \tag{11}$$

$$Z_i = \begin{bmatrix} \cos\alpha_i & \sin\alpha_i \sin\beta_i & \sin\alpha_i \cos\beta_i \\ 0 & \cos\beta_i & -\sin\beta_i \\ -\sin\alpha_i & \cos\alpha_i \sin\beta_i & \cos\alpha_i \cos\beta_i \end{bmatrix} Z_{i-1} \tag{12}$$

The two rotational angles of ith section are α_i and β_i, respectively. They can be determined according to (11)-(12), which are denoted by (13).

$$\begin{aligned} \beta_i &= \arcsin(-y_i) \\ \alpha_i &= \arcsin(x_i / \cos(\beta_i)) \end{aligned} \tag{13}$$

IV. Simulation of the Presented Method

In this section, simulation of the serpentine curve based method is presented. First, the obstacle configuration and the pseudo-code of the presented method are illustrated. Second, the simulation results are shown to validate the effectiveness of the presented method.

A. Obstacle configuration and pseudo-code of the serpentine curve based method

A motion simulation platform is established in the MATLAB platform. All the parameters of the simulated CDSR are the same with its hardware. An obstacle space is showed in Fig. 5, in which *XcYcZc* is the base frame of the robot. The obstacle space is comprised of five boards of which two are Board 1 with the size of 470mm*180mm, and three are Board 2 with the size of 225mm*180mm. The thickness of all boards is 5 mm.

According to the shape and dimensions and of the obstacle space, it can be divided into four segments. Segment ②, ③ and ④ are constraint by the environment. Thus, the curve in these segments can be predefined as time independent arc with constant curvature. Segment ① is unrestricted by the boards. A fixed point is set for the entrance of the obstacles. The coordinate of the entrance point is denoted by (14).

$$\begin{aligned} x_1 &= \int_0^{s_1} \sin\left(\frac{as_1}{2\pi} \sin\frac{2\pi s}{s_1}\right) ds = 0 \\ y_1 &= J_0\left(\frac{as_1}{2\pi}\right) s_1 = h \end{aligned} \tag{14}$$

Where, J_0 is the zero order Bessel function.

The pseudo-code of serpentine curve based motion planning method is presented as follows.

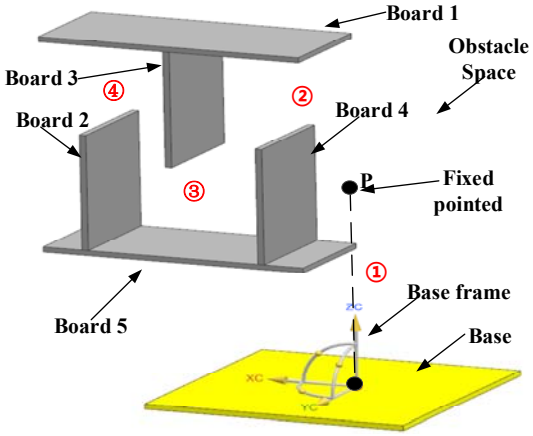

Fig. 5. Parameterization of a point on a spatial curve

Inputs: entrance height h, feed rate arc length S_{feed}, arc length in free space S_{free}, number of points on the curve n_p, number of joints n.

Outputs: joint angles and cable lengths.

For i=1: S_{feed}: S_{free}

Calculate the parameter of the curvature function in free space according to (14).

For j=1: n_p

Calculate the point on the serpentine curve according to (6)-(9).

End

For k=1: n

Find the center point of each joint according to (10).

Calculate angles of all joints according to (11)-(13).

End

Calculate the cable length according to joint angles.

End

B. Simulation result without base motion

The obstacles and the CDSR are simulated in the MATLAB platform. The obstacle avoidance motion can be shown in real-time. The simulation results are shown in Fig. 6, in which Figs. 6 (a)-(d) denotes the motion images of t= 1s, 40s, 60s and 80s respectively.

From Figs. 6 (a)-(d), we can see the joint angles are very large. Simulation data shows the maximum joint angle during the motion is more than 60 degrees. Such configuration will make the stress in the joints very high and deteriorate the stiffness of the robot. For real CDSR, large joint angle is undesirable. As stated in before section, the maximum limit angle of the CDSR presented in this paper is 27 degrees. Without the base motion, the snake robot should bend its arm heavily to avoid obstacles. Thus, the joint angles of each section can be large. However, with slender structure, big angle displacement for many joints snake robot are very difficult. To avoid big displacement of joint, CDSR can be mounted on the third party motion platform such as linear slider, autonomous vehicle and industrial manipulator. With the base motion, the snake robots become more flexible to conduct operation in obstacle space.

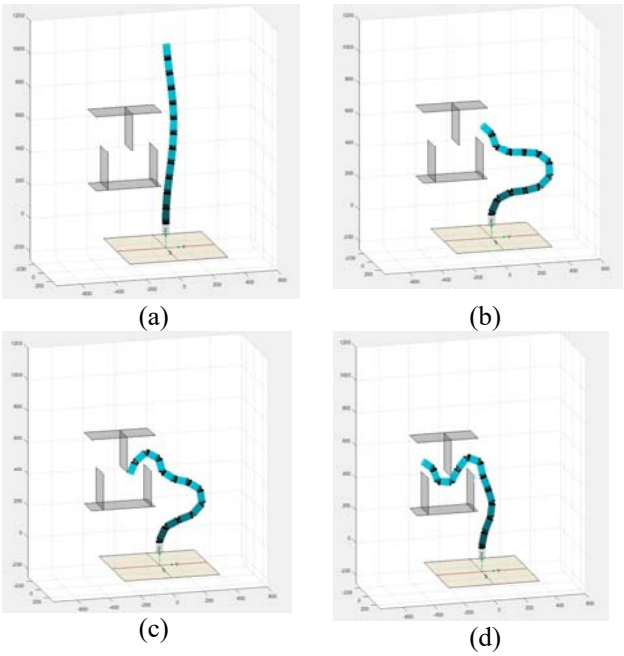

Fig. 6. CDSR motion simulation without base motion.

With the base motion, the joint angles can be reduced greatly. Predefined the serpentine curve along the obstacles, and let the robot follow the curve. Simulation results are shown in Fig. 7, in which Figs. 6 (a)-(d) denotes the motion images of t= 5s, 30s, 40s, and 70s respectively. The maximum joint angle in this situation is 21 degrees.

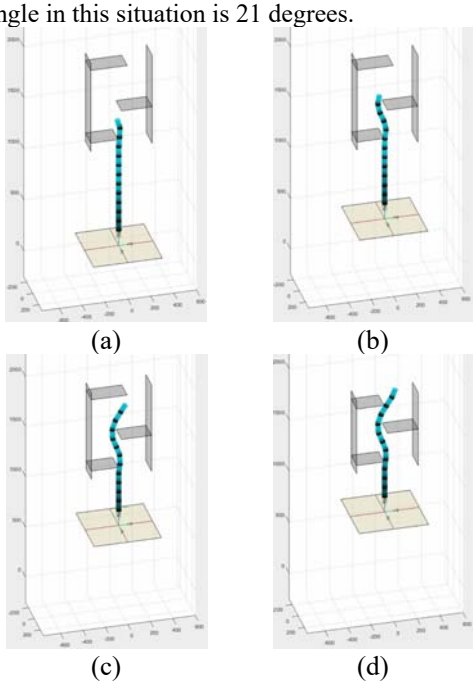

Fig. 7. CDSR motion simulation with base motion.

V. Experiment Validation

Experiments with base motion of the CDSR is performed on the prototype of the robot to validate the presented method. For simplicity, the detailed description the prototype can be referred to [13].

Fig. 8 (a). CDSR motion experiment with base motion at t=5s.

Fig. 8 (a). CDSR motion experiment with base motion at t=40s.

Fig. 8 (c). CDSR motion experiment with base motion at t=70s.

Figs. 8 (a)-(c) are the results of CDSR motion experiment with base motion at t=5s, 40s, and 70s, respectively. Each figure has four sub figures, namely front, left, up and right view of the experiment. The experiment results verify that the CDSR can move around obstacles by the presented serpentine curve based motion planning method effectively.

VI. Conclusion

In this paper, a serpentine curve based method is presented to plan the motion of the CDSR. Firstly, the serpentine curve with curvature functions are used to represent the configuration of the CDSR. Secondly, the curvature of the serpentine curve which satisfies the obstacle constraints is determined. Finally, the CDSR moves along the curve at each control period. The presented method reduces the dimensions of motion planning of the CDSR significantly. Simulation and experiment on a 25 DOFs CDSR prototype validate that

the presented method works well on the designed CDSR platform.

REFERENCES

[1] R. Buckingham *et al.*, "Snake-Arm Robots: A New Approach to Aircraft Assembly," in *SAE Technical Paper*, 2007, pp. 01–3870.

[2] X. Dong *et al.*, "Development of a slender continuum robotic system for on-wing inspection/repair of gas turbine engines," *Robot. Comput.-Integr. Manuf.*, vol. 44, pp. 218–229, Apr. 2017.

[3] R. Buckingham and A. Graham, "Nuclear snake-arm robots," *Ind. Robot Int. J.*, vol. 39, no. 1, pp. 6–11, Jan. 2012.

[4] O. C. Robotics, "Snake-arm robots access the inaccessible," *Nucl. Technol. Int.*, vol. 1, pp. 92–94, 2008.

[5] J. M. Prendergast and M. E. Rentschler, "Towards Autonomous Motion Control in Minimally Invasive Robotic Surgery," *Expert Rev. Med. Devices*, Jul. 2016.

[6] M. Cianchetti *et al.*, "Soft Robotics Technologies to Address Shortcomings in Today's Minimally Invasive Surgery: The STIFF-FLOP Approach," *Soft Robot.*, vol. 1, no. 2, pp. 122–131, Jun. 2014.

[7] Y. Zhou, H. Ren, M. Q.-H. Meng, Z. Tsz Ho Tse, and H. Yu, "Robotics in natural orifice transluminal endoscopic surgery," *J. Mech. Med. Biol.*, vol. 13, no. 02, p. 1350044, Apr. 2013.

[8] M. S. Menon, V. C. Ravi, and A. Ghosal, "Trajectory Planning and Obstacle Avoidance for Hyper-Redundant Serial Robots," *J. Mech. Robot.*, vol. 9, no. 4, pp. 041010-041010–9, 2017.

[9] K. E. Zanganeh and J. Angeles, "The inverse kinematics of hyper-redundant manipulators using splines," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, 1995, vol. 3, pp. 2797–2802.

[10] Z. Zhang, G. Yang, and S. H. Yeo, "Inverse kinematics of modular Cable-driven Snake-like Robots with flexible backbones," in *2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)*, 2011, pp. 41–46.

[11] D. Lau, D. Oetomo, and S. K. Halgamuge, "Generalized Modeling of Multilink Cable-Driven Manipulators With Arbitrary Routing Using the Cable-Routing Matrix," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1102–1113, Oct. 2013.

[12] D. Lau, D. Oetomo, and S. K. Halgamuge, "Inverse Dynamics of Multilink Cable-Driven Manipulators With the Consideration of Joint Interaction Forces and Moments," *Robot. IEEE Trans. On*, vol. 31, no. 2, pp. 479–488, 2015.

[13] L. Tang, J. Wang, Y. Zheng, G. Gu, L. Zhu, and X. Zhu, "Design of a cable-driven hyper-redundant robot with experimental validation," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 5, p. 1729881417734458, Sep. 2017.

[14] Z. Zhang, G. Yang, and S. H. Yeo, "Inverse kinematics of modular Cable-driven Snake-like Robots with flexible backbones," in *2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)*, 2011, pp. 41–46.

[15] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 286–292, Apr. 1995.

[16] G. S. Chirikjian, "Inverse Kinematics of Binary Manipulators Using a Continuum Model," *J. Intell. Robot. Syst.*, vol. 19, no. 1, pp. 5–22, May 1997.

[17] G. S. Chirikjian and J. W. Burdick, "An obstacle avoidance algorithm for hyper-redundant manipulators," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, 1990, pp. 625–631.

[18] G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *Robot. Autom. IEEE Trans. On*, vol. 10, no. 3, pp. 343–354, 1994.

[19] H. Yamada and S. Hirose, "Study on the 3D shape of active cord mechanism," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 2890–2895.

[20] S. Ma, S. Hirose, and H. Yoshinada, "Development of a hyper-redundant multijoint manipulator for maintenance of nuclear reactors," *Adv. Robot.*, vol. 9, no. 3, pp. 281–300, Jan. 1994.

[21] H. Yamada and S. Hirose, "Steering of pedal wave of a snake-like robot by superposition of curvatures," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 419–424.

[22] D. Palmer, S. Cobos-Guzman, and D. Axinte, "Real-time method for tip following navigation of continuum snake arm robots," *Robot. Auton. Syst.*, vol. 62, no. 10, pp. 1478–1485, Oct. 2014.

[23] A. Aristidou and J. Lasenby, "FABRIK: A fast, iterative solver for the Inverse Kinematics problem," *Graph. Models*, vol. 73, no. 5, pp. 243–260, Sep. 2011.

[24]S. B. Andersson, "Discretization of a Continuous Curve," *IEEE Trans. Robot.*, vol. 24, no. 2, pp. 456–461, Apr. 2008.