DOI: 10.17587/mau.22.420-424

D. Yu. Kolpashchikov, dyk1@tpu.ru, **O. M. Gerget**, gerget@tpu.ru,
National Research Tomsk Polytechnic University, Tomsk, 634000, Russian Federation

Corresponding author: **Kolpashchikov D. Yu.,** Engineer,
National Research Tomsk Polytechnic University, Tomsk, 634000, Russian Federation

Accepted on April 22, 2021

Comparison of Inverse Kinematics Algorithms for Multi-Section Continuum Robots

Abstract

Continuum robots are a unique type of robots that move due to the elastic deformation of their own body. Their flexible design allows them to bend at any point along their body, thus making them usable in workspaces with complex geometry and many obstacles. Continuum robots are used in industry for non-destructive testing and in medicine for minimally invasive procedures and examinations. The kinematics of continuum robots consisting of a single bending section are well known, as is the forward kinematics for multi-section continuum robots. There exist efficient algorithms for them. However, the problem of inverse kinematics for multi-section continuum robots is still relevant. The complexity of the inverse kinematics for multi-section continuum robots is quite high due to the nonlinearities of the robots' motion. The article discusses in detail the modification of the FABRIK algorithm proposed by the authors, as well as a Jacobian-based iterative algorithm. A comparison of inverse kinematics algorithms for multi-section continuum robots with constant section length is given and the results of the experiment are described.

Keywords: *forward kinematics, inverse kinematics, continuum robot, algorithm FABRIK*

Acknowledgements: This work was supported by the Russian Foundation for Basic Research no. 20-38-90143 and Russian Federation Governmental Program 'Nauka' no. FFSWW-2020-0014.

For citation:

Kolpashchikov D. Yu., Gerget O. M. Comparison of Inverse Kinematics Algrithms for Multi-Section Continuum Robots, *Mekhatronica, Avtomatizatsiya, Upravlenie*, 2021, vol. 22, no.8, pp. 420—424.

DOI: 10.17587/mau.22.420-424

УДК 004.896

DOI: 10.17587/mau.22.420-424

Д. Ю. Колпащиков, инженер, dyk1@tpu.ru, **О. М. Гергет**, д-р техн. наук, проф., gerget@tpu.ru,
Национальный исследовательский Томский политехнический университет

Сравнение алгоритмов обратной кинематики для многосекционных непрерывных роботов*

Непрерывные роботы — уникальный вид роботов, которые совершают движение за счет упругой деформации собственного тела. Их гибкая конструкция позволяет изгибаться в любой точке тела. Данное преимущество дает возможность использовать таких роботов в рабочих областях со сложной геометрией и множеством препятствий. Кинематика непрерывных роботов, состоящих из одной секции изгиба, достаточно хорошо известна, как и прямая кинематика для многосекционных непрерывных роботов. Однако задача обратной кинематики для многосекционных непрерывных роботов все еще остается актуальной. Сложность задачи обратной кинематики для многосекционных непрерывных роботов является довольно высокой из-за нелинейностей движения робота. В статье подробно рассмотрена модификация алгоритма FABRIK, предложенная авторами, а также итеративный алгоритм, построенный на основе расчета матрицы Якоби. Приведено сравнение алгоритмов обратной кинематики для многосекционных непрерывных роботов постоянной длины и описаны результаты эксперимента.

Ключевые слова: *прямая кинематика, обратная кинематика, непрерывные роботы, алгоритм FABRIK*

Introduction

Continuum robots are flexible manipulators that move due to the elastic deformation of their body. They could be presented as hyper-redundant robots with an infinite number of spherical joints and rigid links between them, where the length of links tends to zero [1]. The ability to bend at any point allows continuum robots to: avoid unwanted collisions in a space with complex geometry and many obstacles; change the direction of movement using contacts with obstac-

*Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-38-90143 "Аспиранты" и государственного задания "Наука" №FFSWW-2020-0014.

les; grab objects with their body. This type of robots is actively used in various fields: industrial robots are used for machining [2], non-destructive testing and repairs inside complex devices [3], [4], in work environments that are difficult to access and dangerous for humans, such as outer space [5] or underwater [6]; medical robots are used as endoscopes and surgical instruments for minimally invasive procedures [7], [8].

Autonomy, real-time motion planning and trajectory optimization of continuum robots are enabled due to inverse kinematics algorithms. Today, a number of approaches are known that solve the inverse kinematics problem of multi-section continuum robots. The geometric approach is simple and effective, e.g., a closed-form solution to the inverse kinematics problem for a multi-section continuum robot by S. Neppalli et al. [9], who determined the length between the beginning and the end of the bending section and thereby determined the bending angle of the section. However, this method offers a solution only for a certain set of configurations. Thus, a complete enumeration of various configurations is carried out while searching for a desired solution.

The inverse kinematics problem can also be solved through an iterative geometric algorithm of Forward And Backward Reaching Inverse Kinematics (FABRIK). The authors in [10] proposed a modification of the FABRIK algorithm for multi-section continuum robots. The modified FABRIK allows considering a continuum robot as a traditional robot with rigid links connected by spherical joints during the forward reaching stage. Then the shape of the continuum robot is restored by the forward kinematics algorithms during the backward reaching stage. This approach is applicable for both serial [10] and telescopic [11] multi-section continuum robots. This algorithm makes it possible to speed up calculations and increase the percentage of correct solutions of high-order accuracy.

Currently, the Jacobian-based methods are widely used to solve the inverse kinematics problem. This approach is successfully applied to various types of multi-section continuum robots: robots with constant [12] and variable curvature [13], as well as concentric tube robots [14]. Among them, Newton's method is used to find the solution iteratively. Iterative approaches based on Jacobian matrices are accurate and capable of working in real time. However, they suffer from high computational complexity and singularity. It should be noted, that testing the algorithm for continuum robots with a large number of bending sections has been of little interest in literature up until this point.

Forward kinematics

This article investigates the forward kinematics that is based on a constant curvature approach, where the bending shape of a continuum robot is treated as an arc of constant curvature. This approach describes the bending of a continuum robot with sufficient accuracy and was verified experimentally [15]. A multi-section continuum robot is defined as a series of curves smoothly flowing into one another. The forward kinematics of a spatial continuum robot is described through transformation matrices:

$$\mathbf{T}_Q = \mathbf{T}_0 \prod_{i=1}^{Q} [\mathbf{T}_\mathbf{Z}(\varphi_i)\mathbf{T}_\mathbf{B}(\theta_i)], \qquad (1)$$

where $\mathbf{T}_0$ — robot's base frame, $\mathbf{T}_\mathbf{Z}(\varphi)$ — rotation around **Z** axis by the rotation angle φ, $\mathbf{T}_\mathbf{B}(\theta)$ — translation along **X** and **Z** axes at the distance depending on the bending angle θ and rotation around the **Y** axis by the bending angle θ, Q — number of bending sections. The transformation matrix $\mathbf{T}_\mathbf{B}(\theta)$ is calculated as:

$$\mathbf{T}_\mathbf{B}(\theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & S(1-\cos(\theta))/\theta \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & S\sin(\theta)/\theta \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where S — bending section length.

FABRIK-based inverse kinematics

The FABRIK algorithm was originally developed to solve the inverse kinematics problem for traditional robots with a finite number of rigid links and joints between them. In order to use FABRIK for multi-section continuum robots, the robot's bending sections are represented as virtual rigid links of variable lengths (chords) that connect the start and end points of each bending section. In the course of solving, forward kinematics are used to reconstruct arcs from the virtual links.

The algorithm iteratively finds a set of bending angles θ and rotation angles φ, so that the coordinates of the robot tip P_Q (a positional component of $\mathbf{T}_Q$) are in the adjacency of the target point t at a distance of no more than the linear tolerance TL, and the angle between the orientation vector of the last section $\mathbf{Z}_Q$ (**Z** axis of $\mathbf{T}_Q$) and the target orientation vector $\mathbf{Z}_t$ of no more than the angular tolerance TA. In addition to the target point t and the target orientation $\mathbf{Z}_t$, the algorithm requires:

1) bending section length S;

2) bending angle limit $\theta_{\max}$.

The algorithm searches for a solution in several iterations. Each iteration includes forward and backward reaching stages. An example of how the algorithm works is shown in Fig. 1.

In the forward reaching stage, the search for a solution is carried out for a robot with rigid links represented as chords with some constant length under no restrictions on the angle of rotation between the chords. At the first step of the forward reaching, the end point P_Q is set to the target point t and then the chain points are redefined starting from the point P_{Q-1} and ending with the point P_1 in accordance with the formulas [16]:

$$P_i^F = (1-\lambda_i)P_{i+1}^F + \lambda_i P_i^B;$$
$$\lambda_i = H_{i+1}/(P_{i+1}^F - P_i^B),$$

where $i \in [(Q-1)...0]$, superscripts F and B denote that the obtained point is a result of forward or backward reaching respectively. The initial points are the backward reaching points.

In the backward reaching stage, the chain points are redefined taking into account the changes introduced by the mutual orientation of the chords. In this stage, the bending (2) and rotation (3) angles of each section are calculated using the inverse kinematics algorithm for a single section:

$$\theta_i = 2\operatorname{sign}(P_{i,X}^F)\arccos\left(\frac{\mathbf{Z}_{i-1}\mathbf{H}_i}{|\mathbf{Z}_{i-1}||\mathbf{H}_i|}\right);$$
$$\theta_i = \begin{cases}\theta_{\max}, \ \theta_i > \theta_{\max}, \\ \theta_i, \quad \theta_i \leqslant \theta_{\max};\end{cases} \tag{2}$$

$$\mathbf{H}_i = \mathbf{P}_i^F - \mathbf{P}_{i-1}^B;$$
$$\varphi_i = \operatorname{arctan2}(P_{i,X}^F, P_{i,Y}^F), \tag{3}$$

where $\mathbf{Z}_i$ is the $\mathbf{Z}$ axis of the *ith* section frame ($\mathbf{Z}_0$ is the $\mathbf{Z}$ axis of the robot's base frame), $\mathbf{H}$ — chord connecting the start and end points of the section, $\theta_{\max}$ — maximum bending angle, $P_{i,x}^F$ — x and $y$components of the P_i^F point in the $\mathbf{T}_i$ frame.

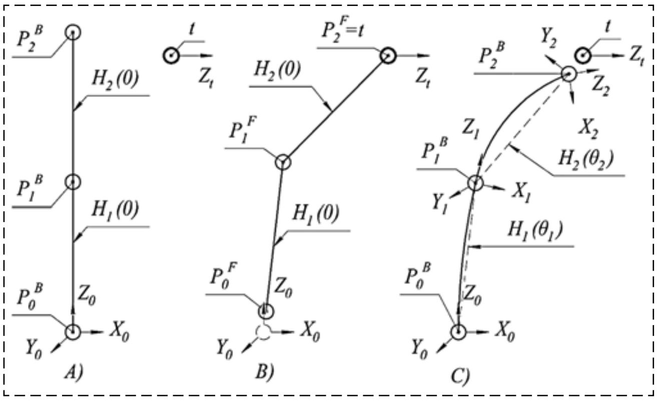

Fig. 1. A complete iteration of the algorithm for a two-section robot (Q = 2).

Note: A — the initial position of the robot, target point and target orientation; B — the result of the forward reaching; C — the result of the backward reaching

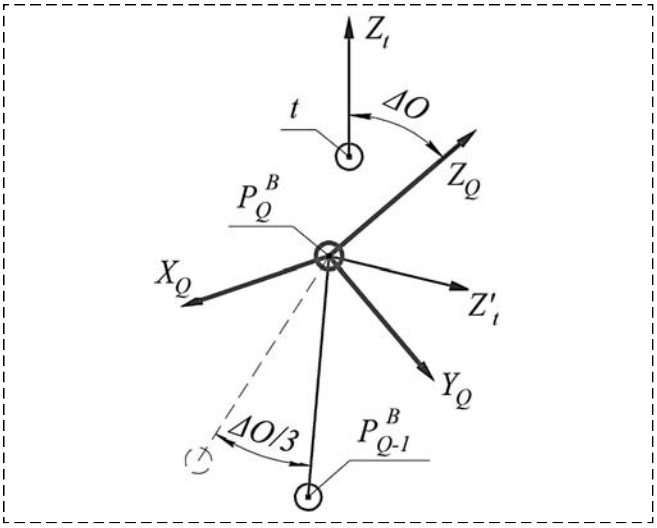

Fig. 2. Tip adjustment scheme

At the next step, the new position of P_i^B and its orientation $\mathbf{Z}_i$ are determined by the formula (1). Thus, all points of the robot from the beginning to the end are redefined. Next, the condition of finding the robot's tip in the adjacency of the target point is checked. If this condition is not met, then the algorithm performs another iteration. Iterations are repeated until the robot's tip reaches the target point.

Upon reaching the target point, the angle between the target vector $\mathbf{Z}_t$ and the orientation vector of the robot's tip $\mathbf{Z}_Q$ is checked. If the angle is greater than the angular tolerance *TA*, then the tip adjustment is carried out (fig. 2). To do this the penultimate point of the robot P_{Q-1}^B rotates around the axis $\mathbf{Z}'_t = \mathbf{Z}_t \times \mathbf{Z}_Q$ with the rotation center P_Q^B by one third of the angle ΔO. ΔO is the angle between the target vector $\mathbf{Z}_t$ and the orientation vector of the robot $\mathbf{Z}_Q$. The rotation is performed by one third of the angle ΔO because vectors $\mathbf{Z}_t$ and $\mathbf{Z}_Q$ have different initial points and it is difficult to predict the robot's behavior after rotating by ΔO. The $\mathbf{Z}_Q$ vector appears closer to the target orientation vector $\mathbf{Z}_t$ in the next iteration.

Besides, rotation around the vector $\mathbf{Z}'_t$ is performed once in several iterations for all points of the robot except for the first and last points to prevent the robot from getting stuck in a certain position.

Jacobian-based inverse kinematics

Jacobian-based approach is the most common way to solve the inverse kinematics problem for both continuum and traditional rigid robots. In this work we use Newton's method, where the Jacobian matrix is evaluated at several iterations:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + (\mathbf{J}(\mathbf{x}_k)^{\mathrm{T}}\mathbf{J}(\mathbf{x}_k) + \mathbf{W}_{\mathrm{sign}})^{-1}\mathbf{J}(\mathbf{x}_k)^{\mathrm{T}}(\mathbf{G} - \mathbf{F}(\mathbf{x}_k))$$

where $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ are current and next robot configuration parameters, $\mathbf{J}(\mathbf{x}_k)$ is the Jacobian matrix for $\mathbf{T}_Q$ with the parameters $\mathbf{x}_k$, $\mathbf{G}$ is the target position and the difference between the target orientation vector $\mathbf{Z}_t$ and the current orientation vector of the robot $\mathbf{Z}_Q$, $\mathbf{F}(\mathbf{x}_k)$ is the current position and the difference between the target orientation vector and the current orientation vector of the robot depending on the configuration parameters $\mathbf{x}_k$, $\mathbf{W}_{\mathrm{sign}}$ is a positive definite diagonal matrix used to avoid singular configurations.

$\mathbf{G}$ is a vector consisting of the coordinates of the target point and 0, which means that the orientation vector of the robot must coincide with the target vector:

$$\mathbf{G} = (X_t\ Y_t\ Z_t\ 0)^{\mathrm{T}}.$$

The current position $\mathbf{F}(\mathbf{x}_k)$ is defined by forward kinematics (1) and the angle between the target and the current vector by the formula:

$$F(x_k) = \begin{pmatrix} P_{Q,x}(\mathbf{x}_k) \\ P_{Q,y}(\mathbf{x}_k) \\ P_{Q,z}(\mathbf{x}_k) \\ \operatorname{acos}(\mathbf{Z}_t^{\mathrm{T}}\mathbf{Z}_Q(\mathbf{x}_k)) \end{pmatrix}.$$

The Jacobian matrix $\mathbf{J}$ contains the linear velocities of the robot's tip $\mathbf{J_P}$ and the rate of change of the angle $\mathbf{J_Z}$ between the target orientation vector $\mathbf{Z}_t$ and the orientation vector of the robot $\mathbf{Z}_Q$.

$$\mathbf{J} = \begin{pmatrix} \mathbf{J_P} \\ \mathbf{J_O} \end{pmatrix}.$$

The movement of each section consists of rotation around the $\mathbf{Z}$ axis and linear plus rotational motions that depend on the same bending angle θ as shown in equation (1). Accordingly, the Jacobian matrix for the linear velocities of a continuum robot from Q sections can be obtained as follows:

$$\mathbf{J_P} = (\mathbf{J}_1\quad \mathbf{J}_2\quad \cdots\quad \mathbf{J_Q});$$

$$\mathbf{J}_i = (\mathbf{J}_i^Z\quad \mathbf{J}_i^B);$$

$$\mathbf{J}_i^Z = \mathbf{Z}_{i-1}(P_Q - P_{i-1});$$

$$\mathbf{J}_i^B = \mathbf{R}'_{i-1}\frac{dP(\theta_i)}{d\theta_i} + \mathbf{Y}'_{i-1}(P_Q - P_i),$$

where R and P are the rotational and positional portions of the transformation matrix $\mathbf{T}$, the upper stroke denotes the matrix $\mathbf{T}_{i-1}$ after rotation $\mathbf{T}_Z$ (see equation (1)).

The Jacobian matrix for the rate of change of the angle between the target orientation vector and the robot's orientation vector is:

$$\mathbf{J_O} = \begin{pmatrix} \dfrac{d\mathbf{F}(\mathbf{x}_k)_4}{dx_1} & \dfrac{d\mathbf{F}(\mathbf{x}_k)_4}{dx_2} & \cdots & \dfrac{d\mathbf{F}(\mathbf{x}_k)_4}{dx_{Q\cdot 2}} \end{pmatrix}.$$

Experiment

The article presents the results of experiments with three-, five- and ten-section continuum robots. To do this 10^6 target points and orientation vectors of the robot's tip were defined using forward kinematics (1) at random values of the bending and rotation angles of the robot's sections. As a result, at least one solution to the inverse kinematics problem was guaranteed for each target point and orientation vector.

Sample generation parameters: length of each section $S = 50$ mm, maximum rotation angle $\varphi_{\max} = 360°$, maximum bending angle $\theta_{\max} = 180°/Q$, where Q — the number of sections in the robot. Rotation of all points around $\mathbf{Z}'_t$ occurred once in 25 iterations. $\mathbf{W}_{\mathrm{sign}} = 0.1$ for the Jacobian-based algorithm. The Jacobian matrix was formed using the built-in MATLAB tools and converted into a function in order to reduce the calculation time.

When the target points and the vectors were set as a result of solving the forward kinematics problem, they were used as an input to the inverse kinematics algorithm. The following limitations were set for the inverse kinematics algorithm: linear accuracy <1 μm; angular accuracy $<10^{-3}$ radians; algorithm running time <50 ms.

The number of target points and orientations that were reached within the set limits formed the main percentage of solutions. An additional percentage of solutions consisted of solutions with a linear accuracy of 1 mm and an angular accuracy of 1 radian. The additional percentage shows if the results of the algorithms can improve when a lower accuracy is set.

Results

This section presents the results of the experiment. The experiment was carried out using MATLAB 2020b on a computer with an Intel Core i7-4790K 4.00 GHz CPU, 16.0 GB of RAM. The results of the experiment are presented in table.

The function $\mathbf{J}(\mathbf{x}_k)$ for a ten-section robot generated in MATLAB took 135 MB of hard disk space. Unfortunately, it was impossible to process a function of this size either for simplicity or for calculations in MATLAB due to the lack of RAM. By comparison, the function $\mathbf{J}(\mathbf{x}_k)$ for a five-section robot calculated in the same way and simplified took only 0.5 MB of hard disk space. Since the angular part of the Jacobian matrix $\mathbf{J_O}$ was the longest (about 34 million symbols in the equation), it was decided to reduce the Jacobian matrix to a positional component. After reducing, the function $\mathbf{J}(\mathbf{x}_k)$ took up to 32 MB

Experiment results

Algorithm	Mean operating time (ms)	Solution percentage		Mean iterations
		Main	Additional	
Three-section continuum robot				
FABRIK	1.9	94.8	96.9	68.2
Jacobian	5.6	94.3	99.1	100.9
Five-section continuum robot				
FABRIK	4.7	95.1	97.6	162.3
Jacobian	15.2	84.2	98.0	26.9
Ten-section continuum robot				
FABRIK	16.4	90.4	96.8	221.9
Jacobian	—	—	—	—

of hard disk space. MATLAB failed to simplify the reduced function $\mathbf{J}(\mathbf{x}_k)$ as it again ran out of RAM. However, MATLAB was capable of calculating the reduced function $\mathbf{J}(\mathbf{x}_k)$One iteration of this function takes around 14 s to run, which exceeds the 50 ms time limit set for the experiment.

Conclusion

The article presented the algorithms that were used for solving the inverse kinematics problem for multi-section continuum robots. It further compared the modified FABRIK algorithm and the Jacobian-based iterative algorithm. The experiments have shown that the FABRIK-based algorithm is able to reach the target point and orientation faster than the Jacobian-based algorithm with similar stability of solutions. At the same time, the efficiency of the Jacobian-based algorithm decreased significantly with an increase in the number of sections: it dropped from 94.3 % of solutions for a three-section robot to 84.2 % of solutions for a five-section robot and failed to solve the problem for a ten-section robot as it required a supercomputer to process information of this volume. An additional percentage of solutions has indicated that an increase in the time limit or less stringent accuracy requirements can equalize both algorithms in stability and even make the Jacobian-based algorithm more efficient for robots with a small number of bending sections. Indicatively, an additional percentage of solutions for a three-section robot was 96.9 % for the FABRIK-based algorithm and 99.1 % for a Jacobian-based algorithm. From the above, we can conclude that the modified FABRIK algorithm is more efficient than the inverse kinematics algorithm based on the Jacobian matrices for systems with a large number of bending sections and high-performance requirements.

References

1. **Walker I. D., H. Choset I. D., Chirikjian G. S.** Snake-Like and Continuum Robots, *Springer Handbook of Robotics*, Cham, Springer International Publishing, 2016, pp. 481—498.

2. **Axinte D.** et al. MiRoR—Miniaturized Robotic Systems for Holistic <italic>In-Situ</italic> Repair and Maintenance Works in Restrained and Hazardous Environments, *IEEE/ASME Trans. Mechatronics*, Apr. 2018, vol. 23, no. 2, pp. 978—981, doi: 10.1109/TMECH.2018.2800285.

3. **Dong X.** et al. Development of a slender continuum robotic system for on-wing inspection/repair of gas turbine engines, *Robot. Comput. Integr. Manuf.*, Apr. 2017, vol. 44, pp. 218—229, doi: 10.1016/j.rcim.2016.09.004.

4. **Buckingham R., Graham A.** Nuclear snake-arm robots, *Ind. Rob.*, 2012, vol. 39, no. 1, pp. 6—11, doi: 10.1108/01439911211192448.

5. **Nahar D., Yanik P. M., Walker I. D.** Robot tendrils: Long, thin continuum robots for inspection in space operations, *2017 IEEE Aerospace Conference*, Mar. 2017, pp. 1—8, doi: 10.1109/AERO.2017.7943940.

6. **Liljeback P., Mills R.** Eelume: A flexible and subsea resident IMR vehicle, *OCEANS 2017 — Aberdeen*, Jun. 2017, vol. 2017-Octob, pp. 1—4, doi: 10.1109/OCEANSE.2017.8084826.

7. **Burgner-Kahrs J., Rucker D. C., Choset H.** Continuum Robots for Medical Applications: A Survey, *IEEE Trans. Robot.*, Dec. 2015, vol. 31, no. 6, pp. 1261—1280, doi: 10.1109/TRO.2015.2489500.

8. **Zhang Y., Lu M.** A review of recent advancements in soft and flexible robots for medical applications, *Int. J. Med. Robot. Comput. Assist. Surg.*, Jun. 2020, vol. 16, no. 3, p. 10.1002/rcs.2096, doi: 10.1002/rcs.2096.

9. **Neppalli S., Csencsits M. A., Jones B. A., Walker I. D.** Closed-Form Inverse Kinematics for Continuum Manipulators," *Adv. Robot.*, Jan. 2009, vol. 23, no. 15, pp. 2077—2091, doi: 10.1163/016918609X12529299964101.

10. **Kolpashchikov D., Laptev N., Danilov V., Skirnevskiy I., Manakov R., Gerget O.** FABRIK-Based Inverse Kinematics For Multi-Section Continuum Robots, 2018.

11. **Kolpashchikov D.** et al. Inverse Kinematics for Steerable Concentric Continuum Robots, *Smart Innovation, Systems and Technologies*, 2020, vol. 154, no. April, pp. 89—100.

12. **Jones B. A., Walker I. D.** Kinematics for multisection continuum robots," *IEEE Trans. Robot.*, Feb. 2006, vol. 22, no. 1, pp. 43—55, doi: 10.1109/TRO.2005.861458.

13. **Mahl T., Hildebrandt A., Sawodny O.** A Variable Curvature Continuum Kinematics for Kinematic Control of the Bionic Handling Assistant, *IEEE Trans. Robot.*, Aug. 2014, vol. 30, no. 4, pp. 935—949, doi: 10.1109/TRO.2014.2314777.

14. **Sears P., Dupont P. E.** Inverse Kinematics of Concentric Tube Steerable Needles, *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Apr. 2007, no. April, pp. 1887—1892, doi: 10.1109/ROBOT.2007.363597.

15. **Webster R. J., Jones B. A.** Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review, *Int. J. Rob. Res.*, Nov. 2010, vol. 29, no. 13, pp. 1661—1683, doi: 10.1177/0278364910368147.

16. **Aristidou A., Lasenby J.** FABRIK: A fast, iterative solver for the Inverse Kinematics problem, *Graph. Models*, Sep. 2011, vol. 73, no. 5, pp. 243—260, doi: 10.1016/j.gmod.2011.05.003.