ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2025, Vol. 86, No. 9, pp. 867–891.

CONTROL IN TECHNICAL SYSTEMS

Development of the Design and Trajectory Planning of a Snake-Like Manipulator Robot Based Empirical Approach

Xinbin Zhang[*,a], Zhuo Chen[,b], Haoxiang Su[**,c], Liguo Tan[*,d], Hongwei Liu[**,e], Manlu Liu[**,f], Jianwen Huo[**,g], and S. V. Novikova[***,h]**

[*] *Harbin Institute of Technology, Harbin, China*
[**] *Southwest University of Science and Technology, Mianyang, China*
[***] *Kazan National Research Technical University named after A.N. Tupolev, Kazan, Russia*
e-mail: [a] xinbinzhang@hit.edu.cn, [b] 986052851@qq.com, [c] 463099610@qq.com, [d] tanliguo@hit.edu.cn, [e] liuhongwei@swust.edu.cn, [f] liumanlu@swust.edu.cn, [g] huojianwen2008@hotmail.com, [h] SVNovikova@kai.ru

Received May 14, 2024
Revised April 22, 2025
Accepted April 29, 2025

Abstract—A new design for a pneumatic flexible manipulator based on a series-parallel composite structure is proposed. The device's flexible skeleton is created following the principle of snake spinal articulations. Adjacent vertebrae are equipped with air reservoir-joints located on two sides, connected in parallel with the spine. The pneumatic joint can be bent left or right in the robot's plane of motion by changing the pressure in the pneumatic chamber. A kinematic model has been developed to calculate the position of the end effector based on the bending angles of the manipulator's joints, which, in turn, are calculated from the pressure in the pneumatic chamber-joints. For the developed design, a method for calculating the planned manipulator trajectory is proposed, based on solving inverse kinematics problems. Several examples of calculating optimal motion trajectories meeting various task requirements are provided. The results of full-scale experiments show that the developed flexible snake-like manipulator accurately follows a pre-planned trajectory and demonstrates extensive movement capabilities on a plane.

Keywords: snake-like manipulator, flexible manipulator, soft manipulator, pneumatic actuator, trajectory planning system synthesis, inverse kinematics problem, empirical approach, movement in confined spaces

DOI: 10.31857/S0005117925090059

1. INTRODUCTION

With the advancement of high-tech technologies, the designs of products and devices are becoming, on the one hand, increasingly complex, and on the other, more compact. As a result, their assembly, maintenance, and other operations often need to be performed in confined spaces of complex configurations. Automating routine maintenance processes for such units requires equipment capable of adapting to the space's topology, avoiding obstacles, and performing service functions without damaging the main structure. Traditional rigid manipulators, such as industrial robotic arms, generally possess the ability to move and satisfy diverse end-effector pose requirements only in open spaces. However, their relatively small number of joints, and consequently, degrees of freedom, significantly limit their capabilities when moving in narrow, confined spaces.

One way to address this issue is to use redundant continuum manipulators, which have the ability to bend while moving and bypass existing obstacles [1–7]. For example, [1] presents a multi-joint cable-driven manipulator for automotive spray painting, enhancing the manipulator's movement

capabilities and its ability to avoid obstacles. In [2, 3], linear or hybrid air hose actuators are used to create a series of flexible manipulators mimicking an elephant's trunk. In [4, 5], a series of flexible manipulators capable of bypassing obstacles and reaching a specific point in a confined space were developed. Continuum manipulators enhance movement capability in three-dimensional space by increasing the degrees of freedom of their joints. However, increasing the degrees of freedom leads to an increase in the size and weight of the drive mechanism due to the use of additional hoses and cables, and an increase in the complexity of kinematic calculations, which in turn increases control complexity and computational costs. All of this collectively limits their practical application. Furthermore, the components of such manipulators are entirely made of rigid materials, which can raise safety concerns when they interact with the environment.

In recent years, thanks to the emergence of new pliable materials, so-called soft manipulators have appeared, capable of easily penetrating complex and narrow spaces due to their ability to actively and passively change their geometric shape. At the same time, their high compliance allows them to safely interact with elements of the surrounding environment, causing minimal damage to both environmental elements and the robots themselves in case of collisions. Soft manipulators have prospects for application in industrial assembly, agricultural product harvesting, minimally invasive surgery, domestic service, and other fields. Soft manipulators often mimic bionic strategies observed in nature. For instance, [8] uses an elephant's trunk imitation, [9] imitates worm movements, in [10–12] the authors imitate the fixation methods of climbing plants, in [13] – the movement of octopus tentacles, and so on. Such manipulators use air cushions, pneumatic muscles, springs, etc., as structural support and are actuated by pneumatic actuators, shape memory alloys, or electroactive polymers [14–17]. The unique properties of soft manipulators are attracting increasing attention from researchers striving to overcome the limitations of existing rigid and flexible manipulator designs [18–27]. Examples of soft manipulators include the OctArm continuum manipulator [18], which is driven by pneumatic muscles and mounted on a tracked mobile robot. In [19], a soft manipulator using a honeycomb pneumatic network was developed, which provides stable motion control and high payload capacity. In [20], the authors developed a soft manipulator using artificial pneumatic muscles that mimics the movement of an elephant's trunk and possesses flexible spatial movement capabilities. In [21], a fluidically actuated soft manipulator is described, and the authors of [22] assembled a multi-joint soft manipulator that uses omnidirectional bending and twisting modules for efficient spatial movement. In [23], a soft manipulator for minimally invasive surgery is proposed, where omnidirectional bending motion is achieved by three pneumatic modules. In [24], a self-organizing soft manipulator based on biological mechanisms such as neurons, pollen tubes, and thick-walled tissue cells was developed. According to the authors, the self-organizing manipulator is capable of changing shape depending on environmental conditions. In [25], a three-joint soft manipulator suitable for underwater operation and allowing precise object gripping using an anti-bending and stretching mechanism was developed. In [26], a modular, hydraulically actuated soft robotic wrist consisting of rotating and bending modules is proposed, capable of performing delicate and precise manipulations in complex deep-sea environments. In [27], a tendon-driven soft manipulator using three modules based on Kresling patterns was developed, which can switch between two stable states: flexible joints with low bending stiffness and rigid links with high stiffness.

Despite numerous studies on soft manipulators, these works are still in their infancy, and their practical application is associated with many difficulties. One problem is that the low stiffness of soft robotic systems reduces their controllability and motion stability under the action of gravitational and inertial forces, although their easily changeable geometry and high plasticity allow them to adapt to complex and narrow spaces. One approach to solving this problem is the development of variable stiffness mechanisms, which can be achieved through the use of materials with controllable stiffness, such as low-melting point alloys [28], or jamming mechanisms [29, 30]. Another approach

involves the incorporation of flexible structures made from rigid materials to enhance the overall stiffness of the robot [31]. The first approach allows for changing the robot's plasticity and geometric shape by adjusting stiffness, but the use of variable stiffness mechanisms affects not only the inherent flexibility of the manipulator but also significantly complicates the design and control. With the second approach, some of the soft manipulator's capabilities for changing geometry will be lost. This article implements the second approach to improve the controllability and stability of soft manipulators. A planar flexible snake-like soft manipulator, inspired by the structure of a snake's body, has been developed for movement in narrow spaces with obstacles. By using a rigid-flexible composite structure for the supporting frame (spine), the manipulator demonstrates a good ability to change its geometric shape with simultaneous motion stability and ease of control. For the designed snake-like manipulator, it is necessary to develop a trajectory calculation algorithm that corresponds to the specific movements of the proposed design. Currently, the most widespread approach to manipulator trajectory planning is based on solving the inverse kinematics problem. This approach is applied to both rigid [32–34] and soft multi-joint manipulators [21, 35]. The inverse kinematics problem is solved using various strategies: in [33], trajectory calculation relies on solving kinematic differential equations, while in [32], an analytical solution is proposed without using derivatives. Heuristic approaches are gaining widespread use today. For instance, in the FABRIK (Forward and Backward Reaching Inverse Kinematics) algorithm [36], the coordinates of a multi-joint manipulator's nodes are calculated by repeated linear shifts towards the target position of the end effector. Trajectory calculation algorithms without using inverse kinematics are also described in the literature. For example, [37] uses a specific modification of a genetic algorithm for trajectory calculation, in [38] the authors propose using fuzzy logic for trajectory planning, and in [39] – hybrid neuro-fuzzy networks. It should be noted that such approaches generally require more computational resources and are more complex to implement than algorithms based on solving inverse kinematic problems.

The trajectory calculation algorithm proposed in this article is based on the inverse kinematic approach. The kinematic model of robot motion is built on a combination of theory and experiments. The forward kinematics problem of the flexible manipulator is formulated based on the results of previously conducted full-scale experiments, which allows the algorithm to be adapted to a specific manipulator with high accuracy. The solution to the inverse problem is obtained analytically. Full-scale experiments were conducted to verify the mobility of the flexible manipulator with the proposed trajectory planning algorithm, which confirmed its high effectiveness in bypassing obstacles.

2. DESIGN AND ASSEMBLY OF THE FLEXIBLE ROBOT

The wide range of motion and maneuverability of snakes in complex, confined environments can be attributed to their unique body structure. In general, a snake's body is a composite structure consisting of both rigid and flexible elements. The central spine provides robust support for the body and is covered by soft muscular tissue. A snake's spine comprises hundreds of vertebrae arranged sequentially. By manipulating muscles in various directions, adjacent vertebrae can achieve horizontal deflections of up to 10 degrees and vertical deflections of up to 3 degrees [40]. Although the bending angle of a single joint is limited, a snake can achieve significant overall body bends by accumulating numerous small bends across individual joints. Based on this, the manipulator should have an analogous rigid-flexible composite structure, consisting of a planar flexible spine and actuated air reservoirs. The entire snake-like manipulator is divided into six flexible sector-joints for independent control, with each flexible joint sequentially connected to the others, as shown in Fig. 1. Each sector-joint, in turn, consists of six serially connected small air reservoir-vertebrae. The vertebrae are joined into a joint using flat flexible hinges, allowing the sector to bend within

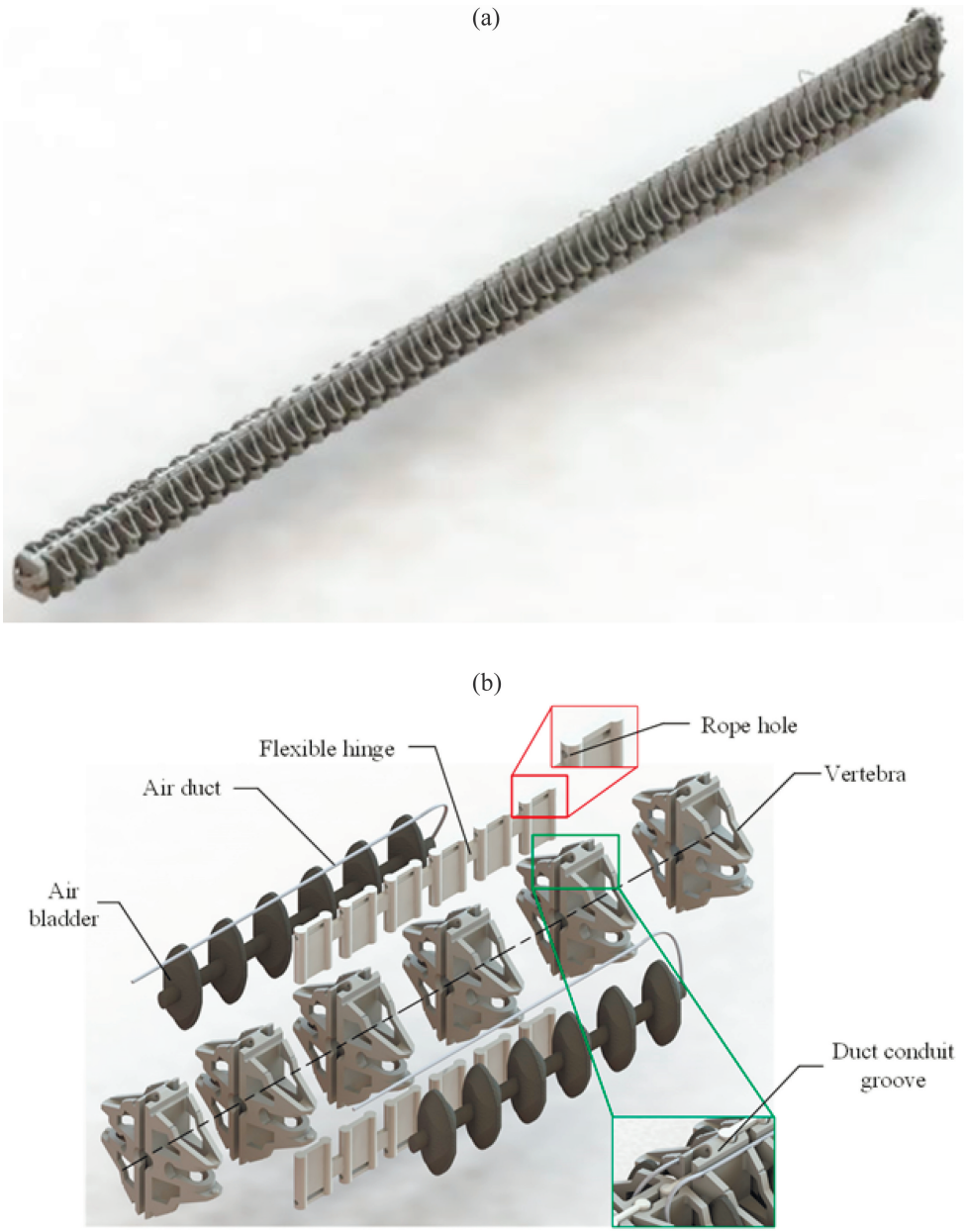


Fig. 1. Flexible manipulator in the shape of a snake: (a) overall view, (b) structure of the flexible joint.

the plane of the joint while restricting longitudinal bending. Additionally, high-strength cables are attached to the outer side of the hinge. Their number increases from tail to head, thereby enhancing the overall longitudinal strength of the flexible joints. Between the vertebrae, on both sides, two pneumatic chambers act as muscles for movement. Each pneumatic chamber contains six internal air reservoirs, corresponding to the operation of the six vertebrae. By changing the air pressure inside the pneumatic chambers on both sides of the structure, they expand and lengthen, thereby regulating the bending angle of the entire flexible joint. To simplify the design, except for the final flexible joint, the lower ("ventral") and upper ("dorsal") surfaces of the vertebrae are equipped with grooves. The depth of the grooves increases as the robot moves from head to tail, resulting in the manipulator acquiring a conical bio-inspired shape resembling an octopus tentacle. To reduce the robot's weight and cost, its spine and flexible hinges were manufactured using 3D printing with

polylactic acid (PLA) and thermoplastic polyurethane (TPU), respectively. The air reservoirs were fabricated using silicone rubber (E640), known for its high elasticity, tear strength, and elongation up to 470%. The air reservoirs were produced by silicone molding. The mold for shaping the reservoirs consists of two parts. Casting is performed by inserting an iron rod into the upper cavity to create the two halves of the reservoir. A vacuum drying oven creates reduced pressure to prevent bubble formation inside the molded reservoirs during their manufacturing process. The finished snake-like manipulator consists of 6 sector-joints. The total length of the robot is 900 mm, and its weight is approximately 2 kg.

3. TRAJECTORY CONSTRUCTION OF TRANSLATING THE SNAKE-LIKE MANIPULATOR BASED ON EMPIRICAL APPROACH

In this section a model is being developed that allows control over the position of the manipulator's end-effector (working tool) in two-dimensional space. We will assume that the end-effector is located at the "head" of the robot. The forward kinematic model of the flexible snake-like manipulator includes two main components: the dependence of the end-effector's position on the bending angle of each flexible joint, and the dependence of the bending angles of the mechanical joints on the pressure in the controlling pneumatic chambers.

3.1. Dependence of the Manipulator's End-Effector Position on Joint Bending Angles

Figure 2 illustrates the motion model of a single sector-joint of the flexible snake-like manipulator. This represents a simplified model with constant curvature, resulting in a smooth continuous curve. The shape of the curve is determined by two variables related to the flexible joint: the length of the joint (L) and the bending angle (θ_i). Based on the kinematic model presented in Fig. 2, the transformation matrix from the initial coordinate system $P_{i-1}(x_{i-1}, y_{i-1})$ to the final coordinate system $P_i(x_i, y_i)$ can be determined. The matrix will thus represent a kinematic mapping from the joint space of the robot to the robot's position space in the external environment. The initial coordinate system $P_{i-1}(x_{i-1}, y_{i-1})$ is transformed into $P_i(x_i, y_i)$ by a single translation and a single rotation:

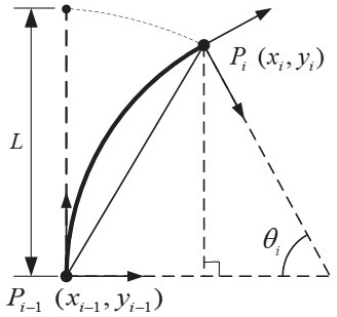


Fig. 2. Bending of a single flexible joint.

(1) The coordinate system $P_{i-1}(x_{i-1}, y_{i-1})$ is translated along two coordinate axes to move point P_{i-1} to point P_i. Based on geometric relationships, we obtain the translation matrix:

$$\mathrm{Trans}\left(L\frac{1-\cos\theta_i}{\theta_i}, L\frac{\sin\theta_i}{\theta_i}\right) = \begin{pmatrix} 1 & 0 & L\dfrac{1-\cos\theta_i}{\theta_i} \\ 0 & 1 & L\dfrac{\sin\theta_i}{\theta_i} \\ 0 & 0 & 1 \end{pmatrix}. \tag{1}$$

(2) The coordinate system rotates around the z-axis by an angle θ_i. The rotation matrix is (2).

$$\mathrm{Rot}\,(z_i,\theta_i)=\begin{pmatrix}\cos\theta_i & \sin\theta_i & 0\\ -\sin\theta_i & \cos\theta_i & 0\\ 0 & 0 & 1\end{pmatrix}. \tag{2}$$

Multiplying these two matrices, we get the transformation matrix from the initial to the final coordinate system for a single joint:

$${}^{i-1}T_i = Trans\left(\frac{L}{\theta_i}(1-\cos\theta_i),\frac{L}{\theta_i}\sin\theta_i\right)Rot(z_i,\theta_i)=\begin{bmatrix}\cos\theta_i & -\sin\theta_i & L\dfrac{1-\cos\theta_i}{\theta_i}\\ \sin\theta_i & \cos\theta_i & L\dfrac{\sin\theta_i}{\theta_i}\\ 0 & 0 & 1\end{bmatrix}. \tag{3}$$

Since the manipulator consists of several serially connected joints with a single degree of freedom of bending, the end-effector's position depends on the bending angle of each joint. If all joints use a constant circular arc curvature model, then according to the coordinate transformation formula (3), the homogeneous transformation matrix from the joint coordinate system $P_n(x_n,y_n)$ to the base coordinate system $P_0(x_0,y_0)$ for n segments can be obtained:

$${}^0T_n = {}^0T_1\times{}^1T_2\times\ldots\times{}^{n-2}T_{n-1}\times{}^{n-1}T_n. \tag{4}$$

Furthermore, if the manipulator is mounted on a slider that can move along the y-axis, the homogeneous transformation matrix from the fixed coordinate system to the base coordinate system is as follows:

$${}^bT_0=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & y_b\\ 0 & 0 & 1\end{bmatrix}. \tag{5}$$

In this equation, the parameter y_b represents the distance the slider moves. Multiplying matrix y_b by the transformation matrix ${}^0T_n$ gives the homogeneous transformation matrix from the fixed end of the manipulator to the manipulator's end-effector:

$${}^bT_n = {}^bT_0\times{}^0T_n \quad (n=6). \tag{6}$$

The position and orientation vectors of the end-effector are given by the relations:

$$\begin{cases}P_n = {}^bT_n[0 \quad 0 \quad 1]^T,\\ D_n = {}^bT_n[0 \quad 0 \quad 1]^T - P_n.\end{cases} \tag{7}$$

3.2. Dependence of Joint Bending Angle on Pressure in Controlling Pneumatic Chambers

Under a constant load, there is a unique relationship between the deformation of the controlling pneumatic chambers and the pressure inside them. During operation, the joint's bending angle depends on the following factors: the pressure applied to the air reservoirs on both sides of the spine; the physical characteristics of the reservoirs' expansion; and the structure of the connecting hinges. Different pressures and varying degrees of reservoir expansion in different sector-joints lead to different bending angles. The unique bending characteristics of each joint of the flexible snake-like manipulator were determined through physical experiments. Hysteresis is a property of systems whose instantaneous response to applied stimuli depends, in part, on their current state. Hysteresis is observed when the current state of a system is determined by external conditions not only at that

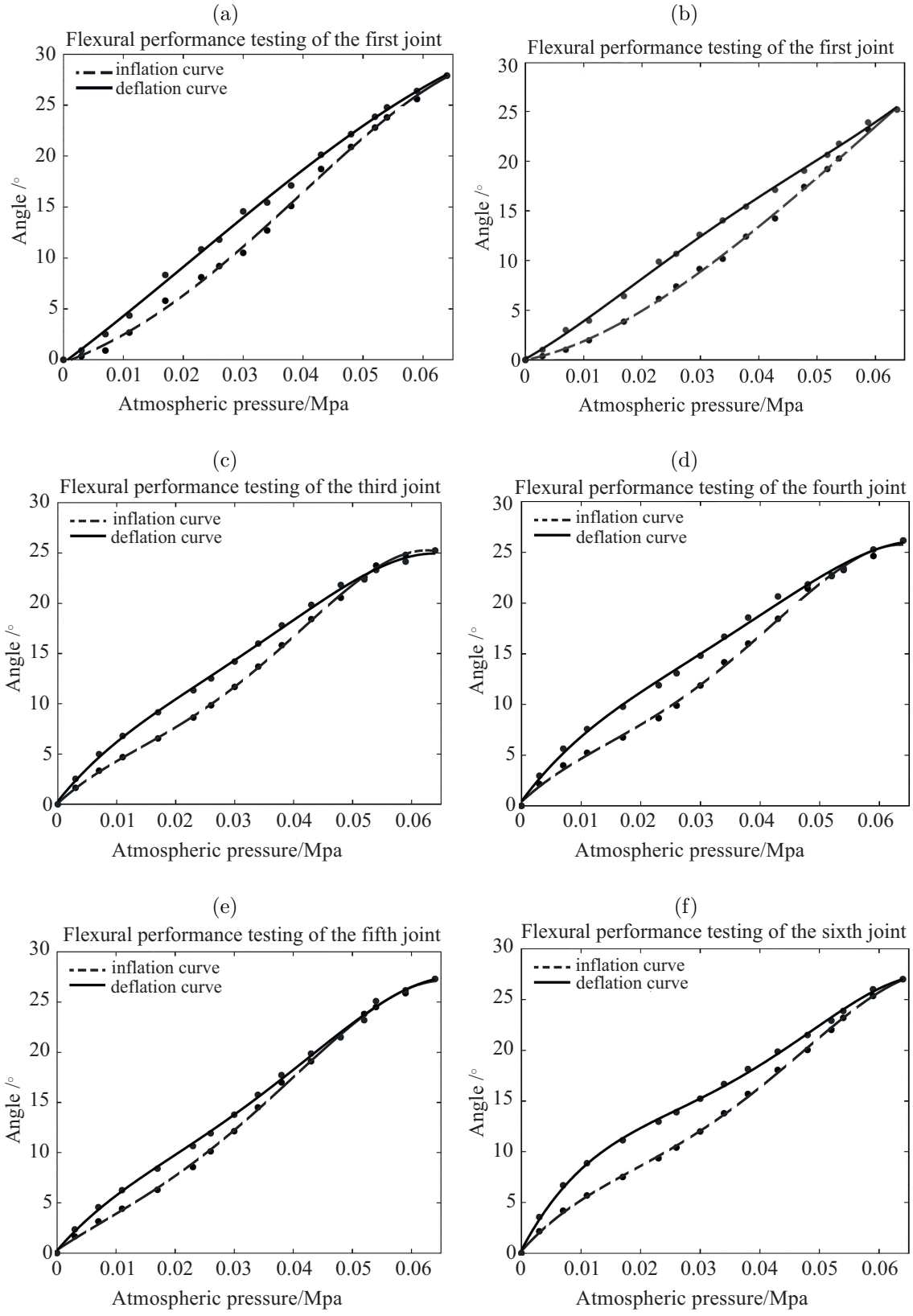


Fig. 3. Mapping of the relationship between joint bending angles and pressure in the controlling pneumatic chamber.

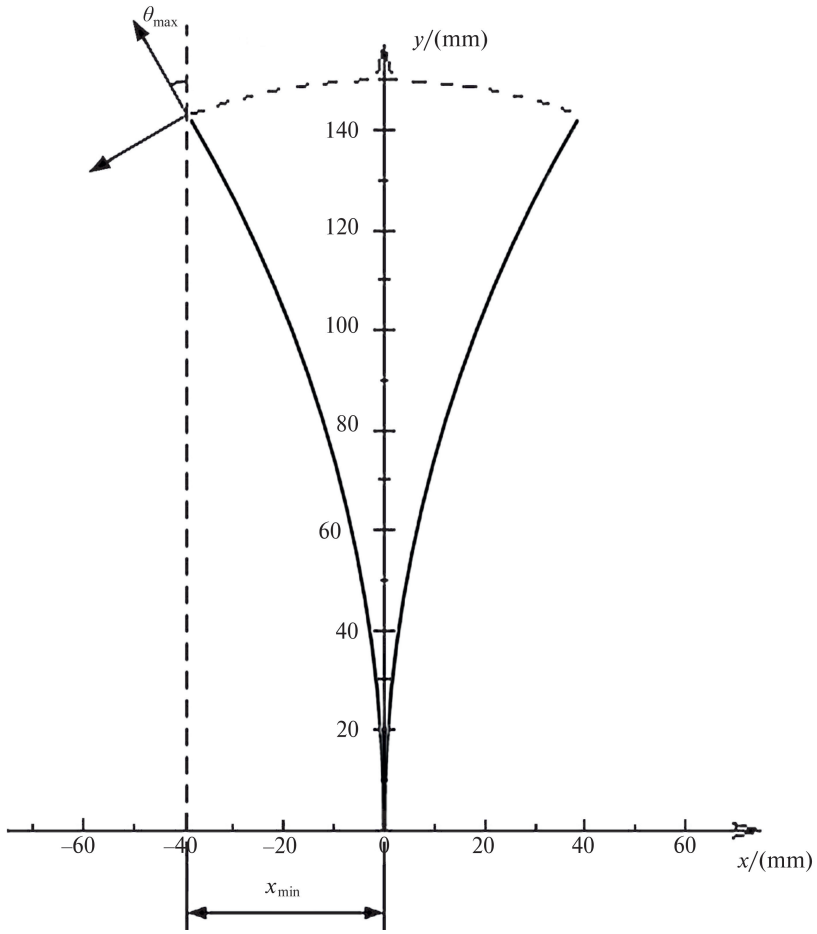


Fig. 4. Reachable range of the end joint.

moment but also at preceding moments in time. For the problem under consideration, this means that the pneumatic chamber's deformation is influenced by its historical states. If two actuated pneumatic chambers are involved in the joint motion, the dependency becomes more complex, and solving for the joint's bending angle becomes difficult. Therefore, the developed flexible manipulator uses a single pneumatic chamber control method, described as follows: When the joint bending angle increases on one side, the pressure in the pneumatic chamber on the opposite side increases. When the joint bending angle decreases on one side, the pneumatic chamber on the opposite side is depressurized and contracts. This means that one pneumatic chamber leads to a unilateral bend of the joint. This control method establishes a direct relationship between the joint's bending angle and the pressure in the controlling pneumatic chamber, which simplifies calculations and facilitates subsequent trajectory planning for the manipulator.

The setup consists of an STM32 microcontroller for regulating the valve, which controls the pressure in the air chambers to induce bending in the snake-like manipulator. A high-precision WT61C bending angle sensor is installed on the first vertebra of each segment to measure the joint's bending angle. Experiments were conducted by inflating the pneumatic chamber on one side while the opposite side remained unloaded. The air pressure range was set from 0 to 0.064 MPa. Several experiments were performed, and the resulting curves were approximated by a fourth-degree polynomial. As a result, "pressure-bending angle" relationships were obtained for all six joints, as shown in Fig. 3. The experimental results with two mirror-imaged pneumatic chambers of the same joint are practically identical; therefore, this article presents results only for one side of the robot. The discrepancy between the approximating "pressure-bending angle" relationship

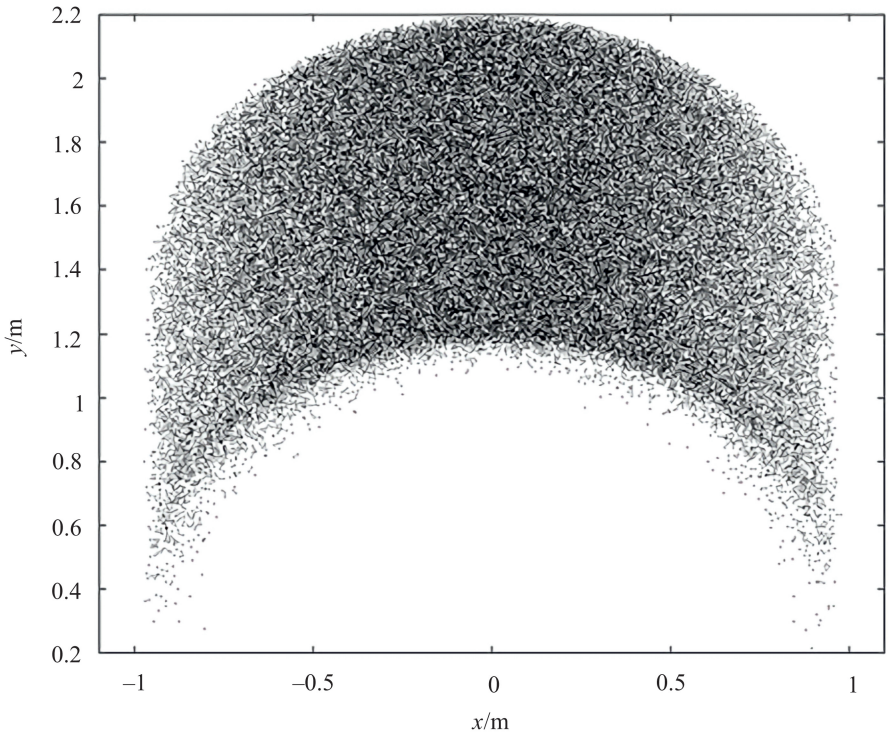


Fig. 5. Workspace of the flexible snake-like manipulator.

and the actual experimental values is less than 5%, reflecting the kinematic characteristics of the joint with high accuracy. The obtained curves serve as the basis for solving the forward and inverse kinematics problems of the robot when calculating the manipulator's end-effector position based on the joint bending angles. As seen in Fig. 3, at the same air pressure levels, the joint's bending angle during the inflation process is smaller compared to the deflation process. The maximum angular lag is about 5 degrees, indicating the pneumatic chamber's hysteresis characteristics. Moreover, due to structural differences between the flexible joints, small variations in corresponding bending angles are observed at the same air pressure, with a maximum deviation of about 3 degrees. Based on this, when solving the forward (inverse) kinematics problem between air pressure and joint bending angle, it is first necessary to determine the air pressure value (or joint bending angle) at the previous articulation, select one of the two experimental fit curves (for inflation or deflation), and then obtain the solution.

The robot's workspace includes the space that can be reached by the base point of the snake-like manipulator's wrist (last link) with its end-effector attached. The workspace is a key criterion for evaluating the manipulator's dynamic performance. Based on the analysis in the previous section, the rotation range of the flexible joints is $[-30°, 30°]$. The reachable range for the final vertebra of the last joint (head) is shown in Fig. 4. To model the workspace of the robot moving on a mobile slider, the Monte Carlo method was applied with a slider movement range of 1000 mm. In each range, 20 000 sets of random numbers were generated to calculate the end-effector's position using the forward kinematics equations. The resulting workspace is demonstrated in Fig. 5.

3.3. Calculating Bending Angles of a Joint Sequence

To solve the inverse kinematics problem for calculating the required bend of each joint, the approach of equal joint bending and linear direction compensation was used. To move the end-effector by an amount x_l in the x-direction (perpendicular to the slider's direction) and an amount y_l in the y-direction (along the slider's direction), the following steps must be performed:

(1) Calculate the required displacement of a single joint in the x-direction by dividing the total displacement by the total number of joints.

(2) Calculate the joint bending angle based on the obtained displacement value along the x-axis.

(3) Calculate the displacement of a single joint along the y-axis when bent by the amount calculated in the previous step.

(4) Calculate the total displacement of the end-effector along the y-axis due to bending by multiplying the displacement of a single joint by their total number.

(5) Calculate the difference between the end-effector's displacement due to bending and the required displacement.

(6) Move the guiding slider by the obtained value, compensating for the lack of robot displacement along the y-axis.

4. TRAJECTORY PLANNING BASED ON EMPIRICAL MODEL

4.1. Planning Based on End-Effector Position Tracking

Given the large number of joints in a pneumatic flexible manipulator, individual trajectory planning with obstacle avoidance for each position of every joint would significantly reduce the efficiency of inverse kinematics calculations. This work adopts a head-tracking strategy, where only the trajectory of the final vertebra of the last joint is tracked. This significantly increases the efficiency of trajectory planning while maintaining the manipulator's flexibility. For the developed flexible snake-like manipulator, the main challenge when using the head-tracking strategy is determining the actual movement trajectory of the end-effector. Essentially, this is the position of the flexible snake-like manipulator upon reaching the operational goal. The following conditions must be met:

(1) The maximum curvature of the trajectory must be less than the curvature corresponding to the maximum joint bending angle.

(2) The guiding slider must be tangent to the trajectory of the manipulator's first joint, and the end of the trajectory must reach the target point.

(3) The total length of the trajectory is an integer multiple of the joint length.

(4) The trajectory must not intersect obstacles.

Based on these requirements, this article considers the construction of end-effector movement trajectories for three different task scenarios.

4.1.1. Reaching a Specified End-Effector Position and Orientation. Based on the modeling and analysis from previous sections, it becomes clear that there is a connection between the end-points of the flexible sector-joints and the joint bending angles. When solving the inverse kinematics problem through path interpolation, there is a possibility of encountering points where no solution exists. To overcome this problem, the article uses a geometric method for finding suitable paths, as shown in Fig. 6. The method consists of the following:

(1) Since the target point and direction of movement are predefined, the target point T can be taken as the starting point, and the direction opposite to the target direction $(-y_e)$ as the initial direction. The path search is performed in reverse, starting from the end-effector and moving towards the robot's base. The direction of reverse movement y (see Fig. 4) is chosen as the direction opposite to y_T (see Fig. 6). This allows for defining the workspace for the end vertebra of the first sector-joint from the head. Within this area, the corresponding position is determined to sketch the first segment of the flexible joint.

(2) The end-point position for the next joint is calculated as the solution to the following empirical optimization problem, consisting of stages A–D:

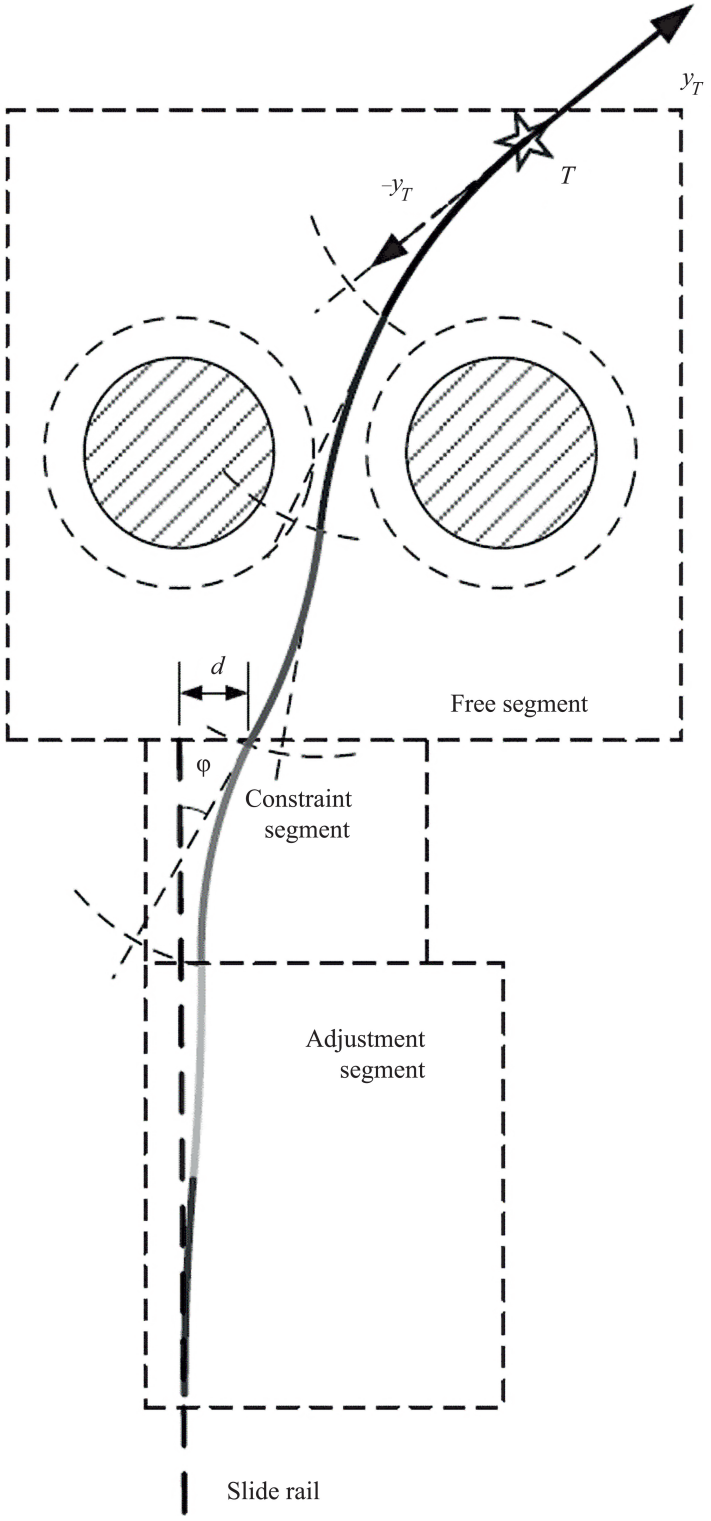


Fig. 6. Robot trajectory search method for a given end-effector position and direction.

(A) The joint's end-point does not intersect obstacles, i.e.:

$$P_1 \notin R_0 \tag{8}$$

Here: R_0 is the forbidden region (set of points of obstacle models). In Fig. 6, examples of forbidden regions are the two circles R1 and R2.

(B) The distance d from the end-point to the guiding slider's line must be minimal.

(C) If, as a result of applying the two above-mentioned principles, the distance d between the end-point and the guiding line is greater than $2x_{\max}$, optimality criterion B changes to D.

(D) The angle φ between the direction of the vertebra's end and the guiding line must be minimal.

(3) The positions of subsequent segment-joints are determined similarly when moving from the end to the base of each joint. The choice of curves for mapping joint positions is made considering approximation to a straight line – the guiding line of the slider's movement without collisions with obstacles. These segments are here called "free segments" (see Fig. 6).

(4) For the subsequent joints to reach the guiding mobile slider, on which the entire robot's base is attached, and for the guiding line to be tangent to the first vertebra of the base joint, the last joint of the free segment must satisfy the following conditions:

(a) The continuation of the last joint's base direction must intersect the guiding line.

(b) The distance d from the base point to the guiding slider's line must not be greater than $x_{\max}$.

(c) The angle φ between the joint's base direction and the slider's guiding line must not be greater than $\theta_{\max}$.

If these conditions cannot be met, it becomes necessary to recalculate the joint angles for the free segment joints. If the requirement for distance d cannot be met, no acceptable solution exists. If the requirement for the deviation angle φ cannot be met, the angle deviation constraint should be introduced into the optimization problem during replanning to ensure that the segment end is sufficiently close to the guiding line while satisfying the maximum angle deviation requirement.

(5) The next section will be called the "limiting segment." It consists of a single flexible joint. Its bending angle is limited by the angle φ, with its base direction parallel to the guiding line.

(6) The last section will be called the "adjustment segment." It includes two joints. By setting the angles of these two joints equal in magnitude but opposite in direction, it's possible to ensure that the first joint's orientation remains unchanged when the distance from the guiding line changes. The bending angle must be chosen to ensure that the first vertebra of the robot's first joint passes tangentially to the guiding line.

At this point, the search for the end-effector's movement trajectory is complete. In this trajectory search method, the critical aspect is using the first three joints for the robot's initial orientation, as this is the minimum number capable of ensuring the correct manipulator position relative to the slider's guiding line. Consequently, only three joints of the free segment are capable of bypassing obstacles. Therefore, under these conditions, the developed flexible snake-like manipulator cannot perform complex obstacle avoidance maneuvers.

4.1.2. Fitting to a Given Trajectory. The method for fitting joint positions to a given trajectory is similar to the previous scenario. For the free segment, the joint ends that intersect the given trajectory are directly selected. The fitting method essentially aligns the free segment's joints along the trajectory. Therefore, the actual trajectory may have a rather large error compared to the planned one. For complex curves, if the end of the free segment does not satisfy the requirements listed above, trajectory fitting may be unattainable.

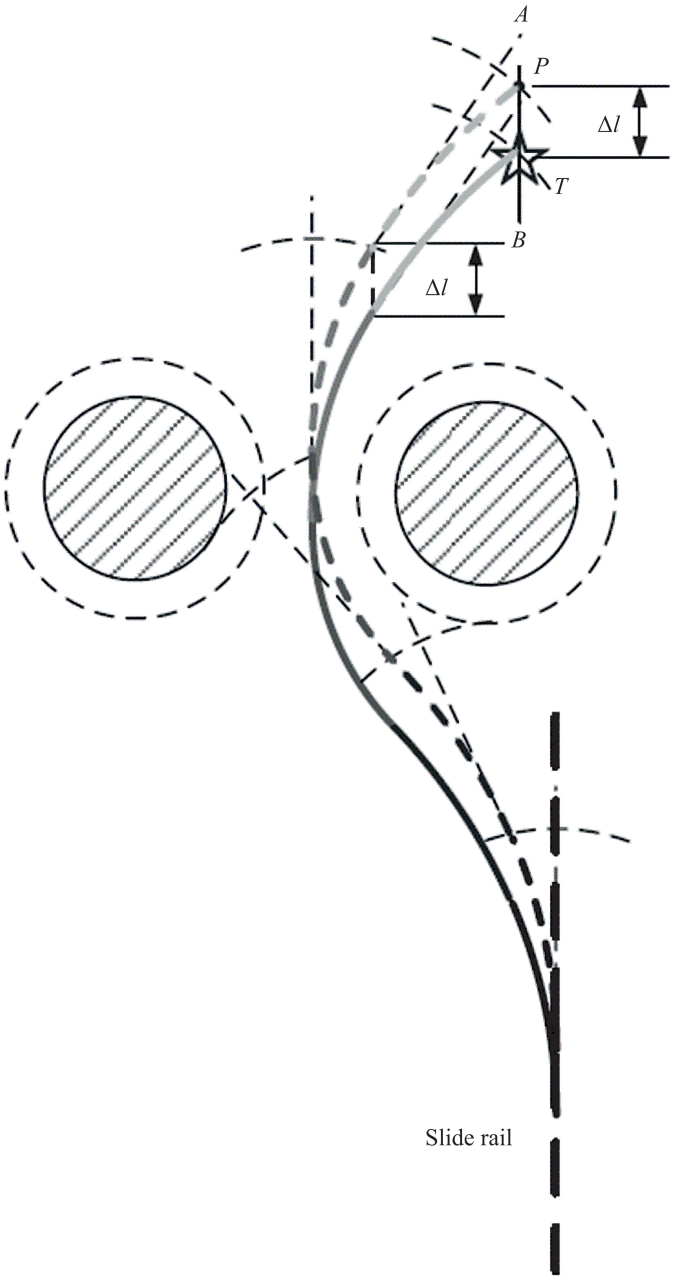


Fig. 7. Robot trajectory search method using permissible regions.

4.1.3. Reaching a Specified End-Effector Position. The trajectory planning methods described above for reaching a target point have limitations in maintaining the manipulator's flexibility and adaptability to the environment. This necessitates the development of a new method for searching the end-effector's trajectory. In accordance with the previously described trajectory planning stages, it is clear that the main problem is achieving a position for the robot's first joint such that the slider's guiding line is tangent to the joint's curve at the robot's attachment point. In many cases, this requires at least three articulations. Given the unfixed orientation of each segment's end, planning to reach a given position, starting from the guiding line, can be done from the base to the segment's end, which increases the number of free joints and simplifies the trajectory planning process.

Let the permissible regions for the base point's location be defined for the base of each segment. Then the planning steps are as follows (see Fig. 7).

(1) Let the target point of the trajectory be denoted by T. A straight line AB is drawn through point T parallel to the slider's guiding line.
(2) At a distance no greater than the joint length L from line AB, a point from the permissible region is selected as the temporary starting point for the last joint's bending trajectory.
(3) The reachable area for the end of the first joint from the temporary starting point is determined.
(4) The position of the joint's end-point in the reachable area is calculated based on constructing the optimization problem A–D, where distance d is replaced by the distance between the end and target points.
(5) Calculations for subsequent joints are repeated similarly, moving towards the end of each joint.
(6) When the joint's end is close to the target point ($d \leqslant 2L$), optimality criterion B changes to D: the angle φ between the final direction and the line connecting the end point and the target point must be minimal.

After the algorithm completes, a bending trajectory for the robot's joints will be obtained, ensuring the achievement of an alternative target point P. The distance between the alternative P and the actual T target points along the line AB is given by Δl. To reach the true target point, the entire trajectory curve needs to be shifted by the amount Δl along the direction AB. If, after the shift, the trajectory intersects obstacle models, it is necessary to plan the trajectory again by selecting different points from the permissible regions at each step as temporary base points for the joints. If there are no intersections, planning is considered complete. This method, compared to previous methods, allows the robotic manipulator to use more joints to directly perform the given tasks, as all joints are utilized to reach the target point and avoid obstacles. Consequently, the flexible robot can effectively perform tasks in more complex conditions, leading to an increased percentage of successfully completed tasks.

4.2. Trajectory Planning Based on Passive Geometric Shape Change

Given the limited force applied for pneumatic chamber deformation, it should be noted that the range of motion of the manipulator's end-effector during active joint bending is often limited. At the same time, the snake-like manipulator possesses flexibility and extensive capabilities for passive geometric shape change. When the sector-joints come into contact with obstacles in the external environment, the manipulator can autonomously bypass these obstacles through its inherent flexibility. This allows the manipulator to adapt to various complex tasks and operating conditions. In the example shown in Fig. 8, during the flexible manipulator's movement, active bending was

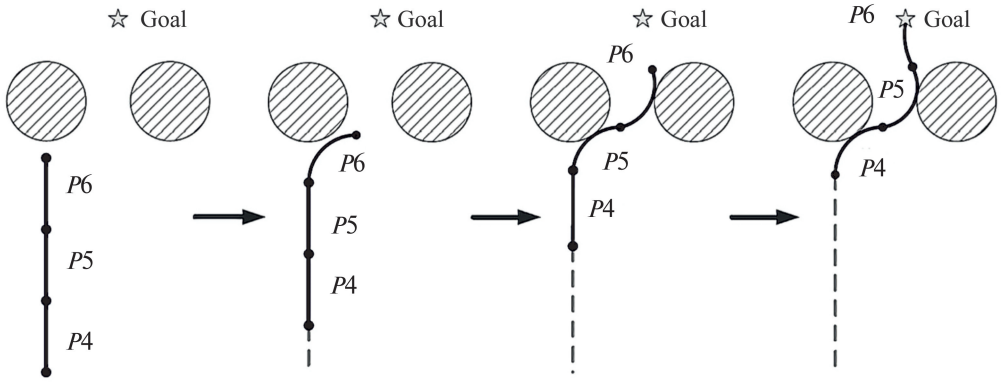


Fig. 8. Trajectory planning based on passive geometric shape change.

only required for the final joint (head). The remaining joints bend upon contact with obstacles and complete their movement solely due to passive bending. The formula for calculating the active bending angle of the last ith joint is:

$$\theta_i = CP_{i-1}L \cdot \arccos\left(\frac{FD_{i-1}}{F \cdot D_{i-1}}\right). \tag{9}$$

where C is the contact point with the obstacle, and F is the tangent vector to the contact surface at the base mark. This strategy effectively utilizes the snake-like manipulator's flexibility, simplifies the trajectory planning process, and consequently, the solution of the given task, but it can lead to frequent collisions with environmental elements. Therefore, this approach is effective in scenarios where the robotic manipulator does not cause damage upon contact with an obstacle or where safety requirements are relatively low.

5. EXPERIMENTS

5.1. Experiment Preparation

The snake-like manipulator is equipped with 12 pneumatic actuating chambers for 6 joints and 12 proportional pneumatic control valves connected to a compressor. The robot is mounted on a linear slider, allowing it to extend and retract by controlling the movement of the slider's linear support. Figure 9 shows the composition of the robotic manipulator's control system. A personal computer (PC) serves as the host for the main control program, written in C language. It communicates with the microcontroller unit (MCU) via a serial communication channel to transmit data. Based on a predefined manipulator motion strategy, the computer calculates the bending angles of each flexible joint at different times and determines the linear guide's displacement. The necessary air pressure values in the 12 control air chambers are determined based on the relationship between air pressure and joint bending angles. The MCU then converts the calculated values into corresponding electrical signals and transmits them to the proportional valves and stepper motors. As a result, the snake-like manipulator performs various movements and obstacle avoidance actions. Two series of experiments were conducted: moving the end-effector along a predefined trajectory without obstacles to investigate its performance, and moving the end-effector to a target point with obstacles.

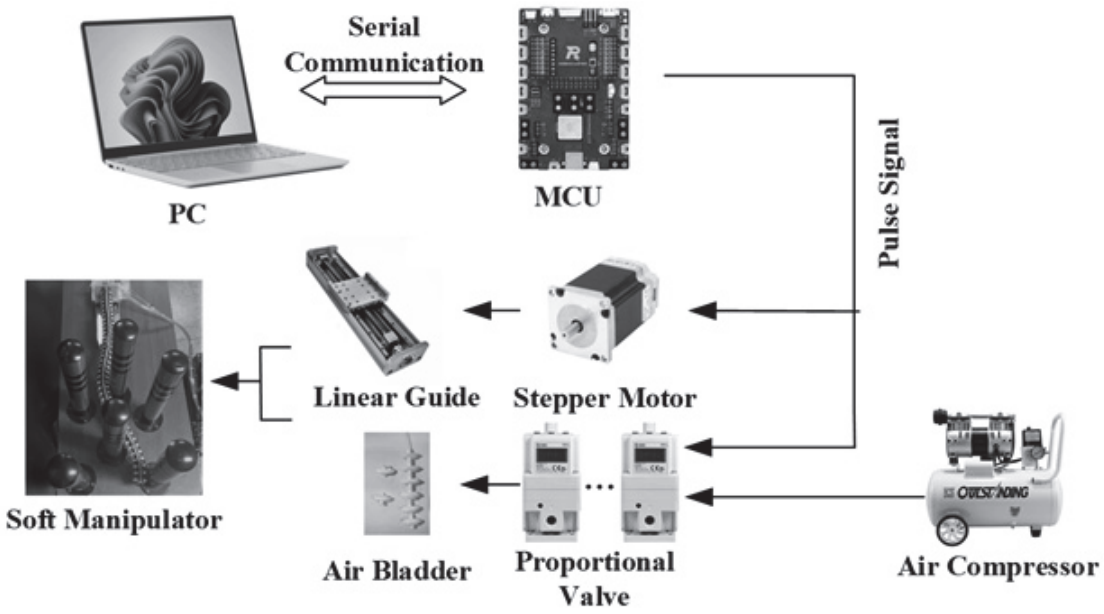


Fig. 9. Schematic diagram of the flexible manipulator's control system layout.

5.2. Experiments on Movement Along a Predefined Trajectory Without Obstacles

This experiment involves creating specific end-effector movement trajectories to test the overall performance of the flexible snake-like manipulator, analyze its movement capabilities, and evaluate its potential application in two-dimensional operations. In these experiments, movement along three trajectories was tested: a straight line, a rectangle, and a circle. A laser is mounted on

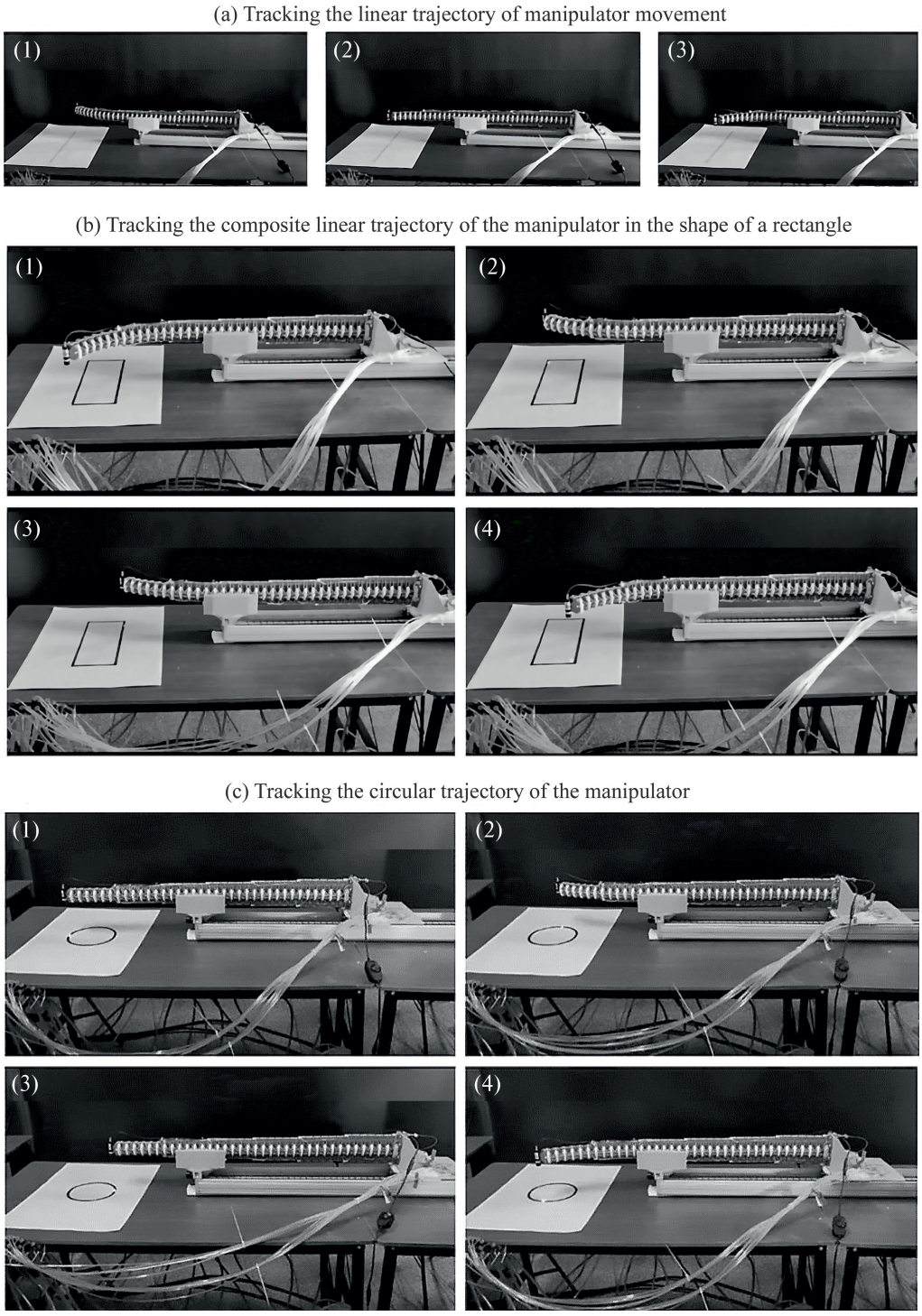
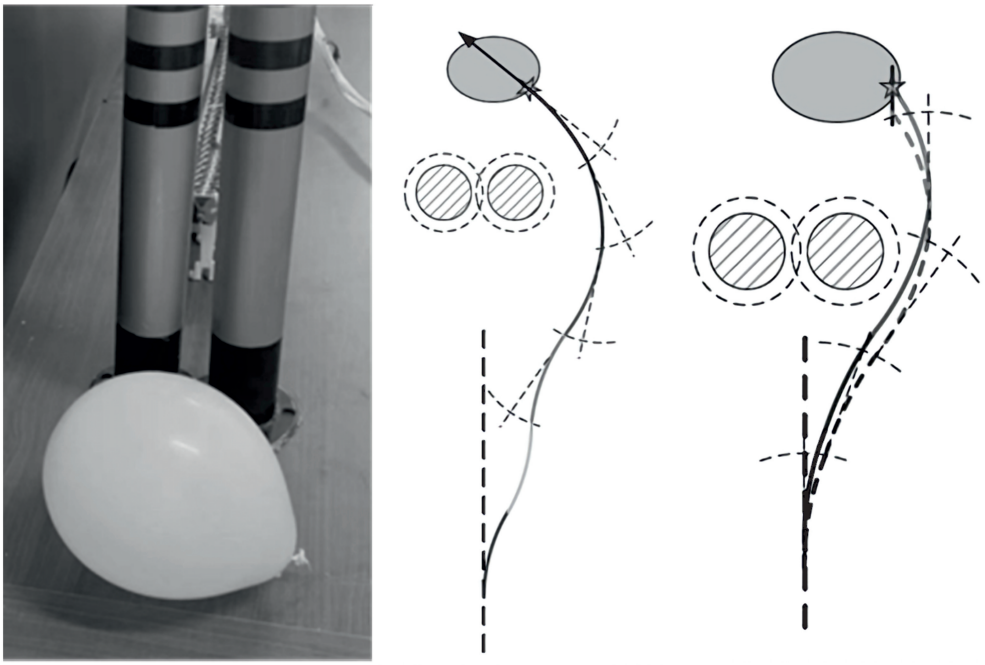


Fig. 10. Experimental results for controlling end-effector movement without obstacles along a given trajectory.

the manipulator's end-joint to track the end-effector's movement trajectory and calculate the error between the actual and predefined trajectories. Figure 10 demonstrates the progress of the three experiments without obstacles: a) tracking a linear trajectory, b) tracking a composite linear trajectory in the shape of a rectangle, and c) tracking a circular trajectory. The error was defined as the deviation of the actual trajectory from the predefined one at the final time point (the end point of a straight line, the corner of a rectangle, and the end point of a circle). As a result, the maximum deviation of the soft manipulator's final position did not exceed 1%, and the deviation of the joint angle in the real experiment from the calculated value did not exceed 2%. This confirms the high effectiveness and accuracy of the flexible snake-like manipulator's movement in two-dimensional space.

5.3. Experiments on Active Obstacle Avoidance Based on End-Effector Position Tracking

The obstacle avoidance experiment, shown in Fig. 11, involves two parallel support pillars as obstacles, and the end-effector is a steel needle attached to the manipulator's last vertebra. The target is a balloon located behind the obstacles. Based on the previously proposed path planning algorithm, the manipulator's end-effector must bypass the obstacles to reach the balloon's position and puncture it. The predefined experimental scenario corresponds to the task scenarios described earlier. Based on the end-effector trajectory planning methods described in previous sections, the trajectories shown in Fig. 11 were constructed. Analyzing the constructed trajectories, it can be concluded that using the method with achieving a specified position and orientation requires a longer guiding slider and involves a greater number of joints. Since there are no strict requirements for the end-effector's position in this experiment, a trajectory corresponding to the method of achieving a specified position was chosen. The speed of the moving slider was set at 25 mm/s, and the time required to bypass the obstacles was set at 20 seconds. To calculate the time-



(a) Experimental scene

(b) End-effector trajectory planning or a given position and orientation

(c) End-effector trajectory planning for a given position

Fig. 11. Experiment on active obstacle avoidance.

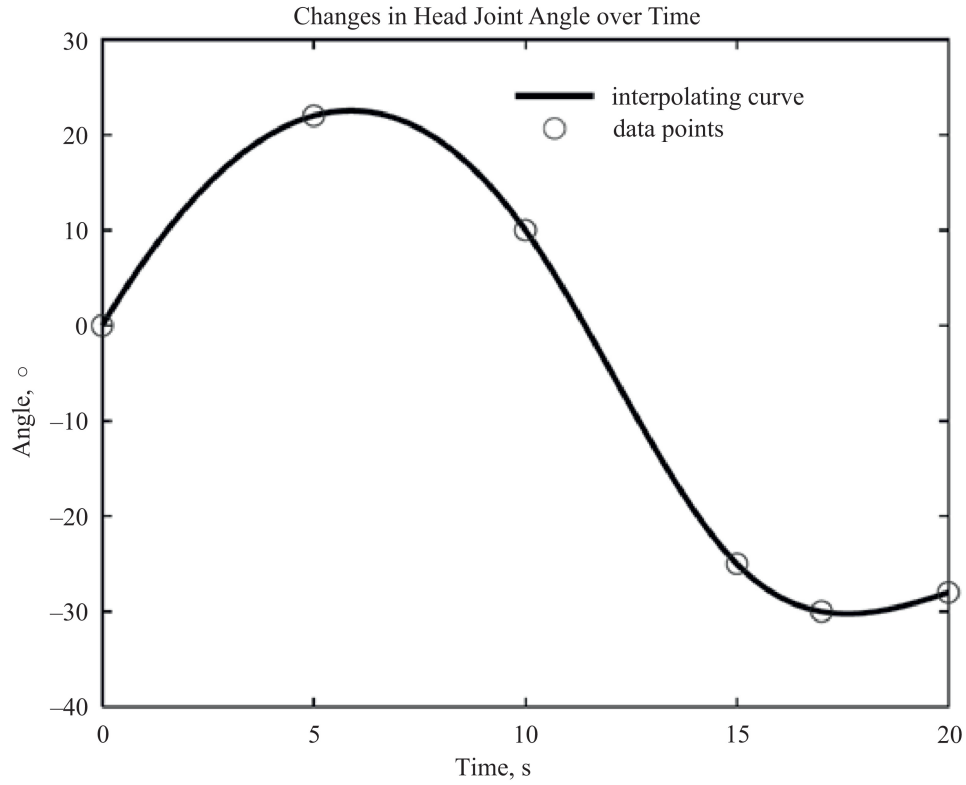


Fig. 12. Calculation of the end-joint angle change trajectory during obstacle circumvention.

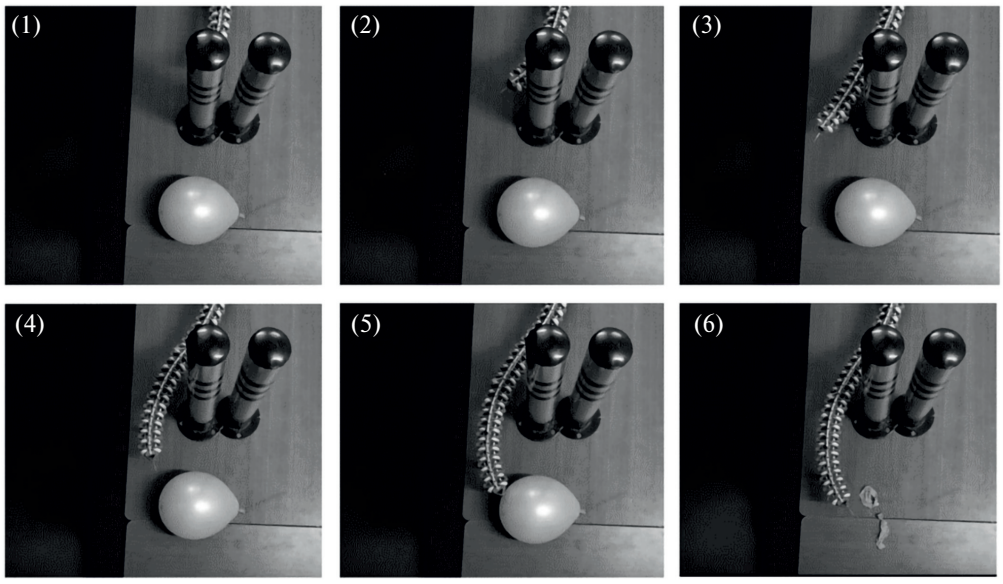


Fig. 13. Experiment on the movement of a flexible snake-like manipulator through active geometric shape change, bypassing a single obstacle.

varying curvature of the end-joint carrying the effector, cubic spline interpolation was applied to four calculated angular positions of the end-effector, as shown in Fig. 12. At the 17-second mark, an additional point of 30° was added to prevent the joint from exceeding its allowable bend. Obstacle avoidance by the flexible manipulator is achieved based on the obtained curve of the end-joint's angle change using the kinematic model and the end-effector tracking strategy. Figure 13

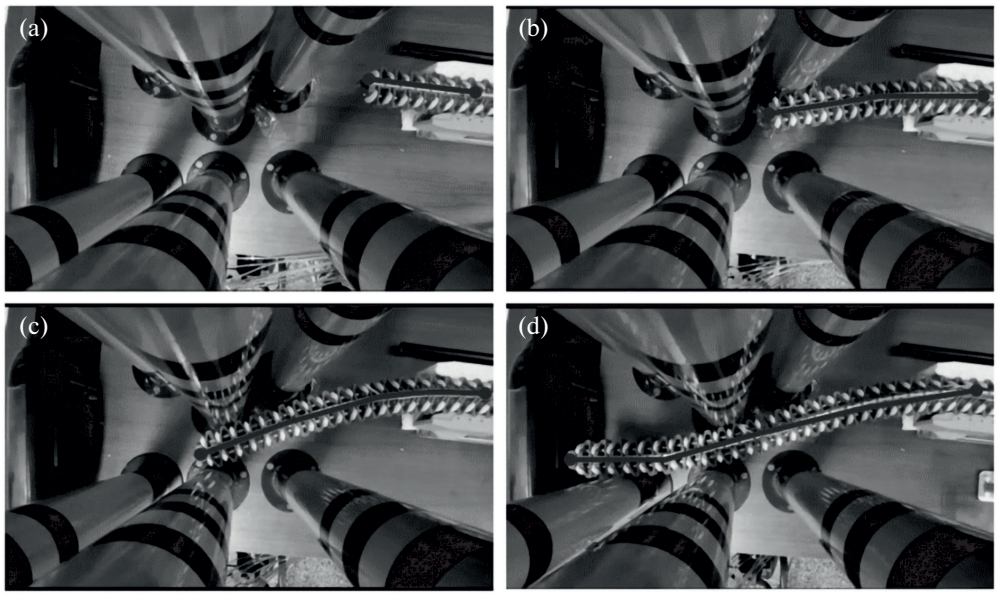


Fig. 14. Experiment on the movement of a flexible snake-like manipulator based on passive geometric shape change, bypassing a group of obstacles.

demonstrates the experimental setup for movement through active geometric shape change of the flexible manipulator during obstacle avoidance. The manipulator is initially positioned in front of the obstacle, moves around the obstacle, bypassing it from the right as it moves, and punctures the balloon located behind the obstacle. It is evident that the snake-like manipulator successfully completed its movement, bypassing the obstacles along the predefined trajectory, with joint angle deviations from the calculated values of no more than 5%.

5.4. Experiment on Obstacle Avoidance Based on Passive Geometric Shape Change

For obstacle avoidance scenarios based on passive geometric shape change, as proposed above, control is only required to calculate the trajectory of the head joint with the effector to ensure the manipulator moves in the desired direction between obstacles. Figure 14 shows the progress of the experiment on soft manipulator movement based on passive geometric shape change to bypass a group of obstacles. Changes in the joint bending angles during movement are also illustrated in Fig. 14. As a result of the experiment, the snake-like manipulator successfully completed its movement, bypassing multiple obstacles and reaching the designated target position, using its active and passive capabilities for geometric change.

6. COMPARATIVE ANALYSIS OF THE PROPOSED APPROACH WITH KNOWN METHODS

This study proposes an original design for a flexible snake-like manipulator and a methodology for constructing its trajectory planning system based on real experiments. To analyze the effectiveness of the developed design and the corresponding trajectory planning system, a comparison of the proposed approach with known designs and methods for calculating flexible manipulator trajectories was conducted.

6.1. Comparison of Designs

A comparison of the features of the developed flexible snake-like manipulator design with known counterparts is presented in Table 1.

Table 1. Characteristics of Flexible Manipulators of Various Designs

Manipulator Type	Construction Type	Actuation Type	Degrees of Freedom	Geometric Shape Change Method	Attachment /Movement Method	Trajectory Calculation Method
Industrial Robotic Arm (KUKA LBR IIWA, Rozum PULSE, etc.)	Constant stiffness (high stiffness)	Mechanical	3–7	Active	On a stationary stand	Trajectory generation based on industrial software
Multi-joint Robotic Arm [1]	Constant stiffness (medium stiffness)	Cable-driven	5–20	Active	On a stationary stand	No information
Super-soft Liquid Elastomer Manipulator [21]	Constant stiffness (low stiffness)	Hydraulic	15	Passive /Active	On a stationary suspension	Trajectory calculation based on position tracking with computer vision
Soft Manipulator based on Jamming [29]	Variable stiffness	Pressure-hardening granulate	–	Active	On a mobile carrier/slider	Trajectory generation based on impulse commands
Proposed Flexible Snake-like Manipulator Model	Variable stiffness	Pneumatic	16	Passive /Active	On a slider	Trajectory calculation based on experimental studies

As Table 1 illustrates, the flexible snake-like manipulator design presented in this work has several advantages over its counterparts, particularly the ability for passive geometric change, variable stiffness, and a large number of degrees of freedom. A key feature is the specific trajectory calculation algorithm, which is developed based on the physical data of the specific snake-like manipulator obtained through a series of experiments.

6.2. Comparison of Trajectory Planning Approaches

Let's consider several approaches for trajectory planning described in modern literature.

- **Trajectory planning for a multi-joint redundant manipulator based on a genetic algorithm [37]:** The manipulator's trajectory is represented by the coordinates of its nodes, with the end node associated with the end-effector. The trajectory must ensure minimal discrepancy between the end-effector's position and the target point. The trajectory is formed as a solution to an optimization problem where the objective function is the deviation of the end-effector's coordinates from the target coordinates, and the constraints are the requirements that the manipulator's links do not intersect obstacles or each other, and that joint rotation angles are permissible. Optimization is performed using a genetic algorithm, where trajectories act as chromosomes and node coordinates as genes. The algorithm is repeated many times to obtain

a "genealogical tree" – a sequence of suitable trajectories that move the manipulator from a given initial position to a final one.

- **Manipulator trajectory calculation based on an analytical solution to the inverse kinematics problem [32]:** The rotation angles of a three-link manipulator (shoulder-forearm-gripper) are determined based on the required position of the last link (gripper). The method involves sequentially formulating and solving systems of nonlinear equations reflecting the dependencies between the rotation angles of subsequent and preceding links. Due to the relatively simple configuration and limitation on the number of links, the problem has a unique solution.
- **Multi-joint manipulator trajectory calculation based on the FABRIK (Forward and Backward Reaching Inverse Kinematics) algorithm [36]:** The trajectory of a multi-joint manipulator is calculated according to repeated forward and backward iterations – passes. The initial node is considered rigidly fixed, and the end-effector is conditionally located at the last node. In the forward pass, the nodes move from the end-effector to the initial node. The last node, with the end-effector, moves to the position specified as the target. Then each subsequent node moves to a position lying on the straight line connecting it and the previous, already shifted, node, at a distance equal to the link length. As a result, after all nodes have shifted sequentially, the initial node will be offset from its original position. Since the initial node is considered fixed and its position cannot change, the initial node's position is corrected – it returns to its initial position. After this, a backward pass is performed – all manipulator nodes are shifted again, but now from the initial to the end node, following the same principle as in the forward pass. Forward and backward passes are repeated until the end-effector is sufficiently close to the given target position.

A comparison of the characteristics of the reviewed methods, as well as the proposed method based on an empirical approach, is presented in Table 2.

Table 2. Features of Trajectory Generation Methods for Multi-joint Manipulators

	Method Group	Algorithm Type	Number of Manipulator Links	Workspace Dimension	Consideration of Specific Manipulator Features	Obstacle Avoidance	Structural Movement
Trajectory calculation based on genetic algorithms	Random Search	Single-pass, iterative	Unlimited	3D	No	Yes	Requires algorithm modification
Analytical trajectory calculation	Analytical Model	Single-pass, non-iterative	2–3	3D	Partially	No	No
Forward and Backward Reaching Inverse Kinematics (FABRIK)	Heuristic Algorithm	Two-pass, iterative	Unlimited	3D	No	Requires algorithm modification	No
Trajectory calculation based on empirical approach	Empirical Algorithm	Two-pass, iterative	Unlimited	2D	Yes	Yes	Yes

The proposed method, unlike other existing trajectory generation methods, is tailored to a specific manipulator, as the calculation is performed based on empirical data obtained through experiments. The analytical method, which also involves calculations for a specific manipulator, is

based on theoretical, but not practically measured, device characteristics. The proposed algorithm is not designed for changing the manipulator's position in three-dimensional space, but it can account for the movement of the initial node due to the mobile slider.

6.3. Comparison of Computational Costs

To compare the proposed algorithm with other manipulator trajectory generation algorithms in terms of computational costs, we will evaluate the computational time complexity for the respective algorithms. Computational time complexity is estimated by a function that expresses the number of required operations as a function of the input data size n. This section presents values of time complexity as $n \to \infty$, i.e., asymptotic computational time complexity [41].

- **Assessment of Genetic Algorithm computational complexity:** With typical choices of transformation methods (point mutation, single-point crossover, roulette wheel selection), the complexity of genetic algorithms is $O(g(nm+nm+n))$, where g is the number of generations, n is the population size, and m is the size of individuals. Thus, the computational complexity of the algorithm can be estimated as $O(n^2)$.
- **Assessment of Analytical Trajectory Calculation Method computational complexity:** The computational complexity of the analytical method is the lowest. Since the algorithm does not involve iterative calculations, the computational volume remains constant, and the complexity is $O(1)$.
- **Assessment of Forward and Backward Reaching Inverse Kinematics (FABRIK) algorithm computational complexity:** The FABRIK algorithm consists of inner and outer loops. In the inner loop, forward and backward passes of the algorithm are sequentially performed. Each such pass computationally generates a decomposition tree of depth $\log_2 n$. Consequently, the

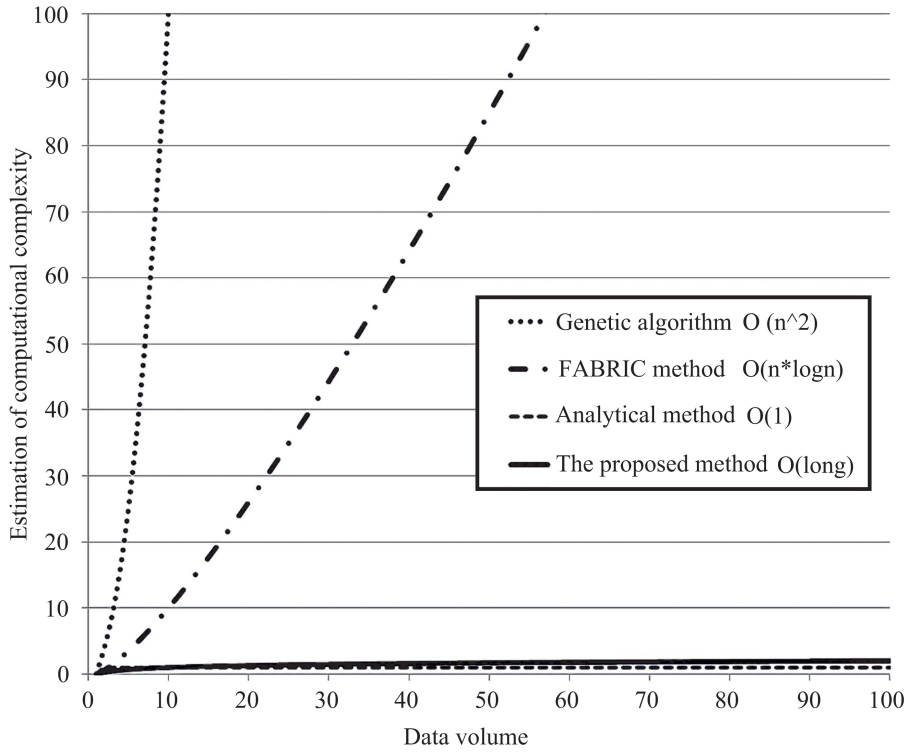


Fig. 15. Growth of computational complexity with increasing data volume for the reviewed trajectory calculation methods of flexible manipulators.

complexity of the inner loop with a double pass is $O(n \log_2 n)$, which for $n \to \infty$ is equivalent to a complexity of $O(n)$. The outer loop of the algorithm repeats the inner loop a number of times proportional to the input data volume n. Therefore, the computational complexity is estimated as $O(n^2)$.

- **Assessment of Trajectory Calculation Algorithm Based on Empirical Approach computational complexity:** The proposed algorithm, unlike the FABRIK algorithm discussed above, consists of a single loop. There is no outer loop, as the end link is located on a slider and can change its position. Consequently, the complexity of the proposed algorithm is $O(n)$.

A graphical comparison of the computational complexity estimates is presented in Fig. 15.

7. CONCLUSION

This article, building upon international research in soft robotic manipulators, presents the design and development of a 6-joint, 13-degree-of-freedom snake-like flexible manipulator. Its design is based on the anatomical structure of a snake's spine. The overall configuration features a series-parallel composite structure, controlled by 12 sets of pneumatic chambers on both sides of the manipulator and a linear guide-slider. We performed a kinematic mapping from the manipulator's joint space to its workspace using a constant curvature model. Through experiments, the relationship between air pressure in the pneumatic chambers and the joint bending angle was established. A unilateral movement method along a linear slider was employed to model the robot's forward kinematics. To meet the requirements of practical engineering applications for a flexible manipulator, we developed trajectory planning methods for obstacle avoidance. Two methods for calculating the robot's movement trajectory were proposed based on the end-effector position tracking approach. We also analyzed a method for obstacle avoidance based on the passive geometric shape change of the soft robot. The conducted physical experiments confirmed the feasibility and correctness of the developed methods. A comparison of the proposed flexible manipulator design approach with existing design solutions demonstrated the originality of the developed structure. A key advantage of the designed snake-like flexible manipulator is its ability to passively change its geometric shape during movement for obstacle avoidance, as well as its variable stiffness with a large number of degrees of freedom. The manipulator is designed for movement and operation on a planar surface, making its application preferable in a range of practical tasks. The algorithm developed for calculating the manipulator's trajectory is based on experimental measurements of the characteristics of the pneumatic chamber-joints, which enable the bending and straightening of the flexible snake-like manipulator. This algorithm has low computational complexity, surpassed only by analytical modeling, which, in turn, cannot offer broad trajectory modification capabilities. Future research plans include further expansion of studies on the design, trajectory planning, and control of the flexible snake-like manipulator. Specifically, we will investigate control methods that allow for variable stiffness of the manipulator. Additionally, the manipulator may be equipped with multiple integrated sensors for real-time position tracking, which will enhance the robot's response speed and control accuracy.

REFERENCES

1. Larson, O. and Davidson, C., Flexible arm, particularly a robot arm, U.S. Patent No. 4,393,728, 19 Jul. 1983.
2. Hannan, M.W. and Walker, I.D., Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots, *J. Field Robot.*, 2003, vol. 20, no. 2, pp. 45–63.
3. McMahan W., Jones B., and Walker I. D. Design and implementation of a multi-section continuum robot: Air-Octor. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005. P. 2578–2585.

4. Robert, B., Robots in the nuclear industry: a review of technologies and applications, *Ind. Robot.*, 2011, vol. 38, no. 2, pp. 113–118.

5. Rob, B. and Andrew, G., Nuclear snake-arm robots, *Ind. Robot.*, 2012, vol. 39, no. 1, pp. 6–11.

6. Camarillo, D.B., Milne, C.F., Carlson, C.R., et al., Mechanics modeling of tendon-driven continuum manipulators, *IEEE Trans. Rob.*, 2008, vol. 24, no. 6, pp. 1262–1273.

7. Xu, W., Liu, T., and Li, Y., Kinematics, dynamics, and control of a cable-driven hyper-redundant manipulator, *IEEE/ASME Trans. Mechatronics*, 2018, vol. 23, no. 4, pp. 1693–1704.

8. Rolf, M. and Steil, J., Constant curvature continuum kinematics as fast approximate model for the bionic handling assistant, *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 3440–3446.

9. Yang, G.Z., Bellingham, J., Dupont, P.E., et al., The grand challenges of Science Robotics, *Sci. Robot.*, 2018, vol. 3, no. 14, pp. eaar7650.

10. Dumais, J. and Forterre, Y., Vegetable dynamicks: the role of water in plant movements, *Annu. Rev. Fluid Mech.*, 2012, vol. 44, pp. 453–478.

11. Argiolas, A., Puleo, G.L., Sinibaldi, E., et al., Osmolyte cooperation affects turgor dynamics in plants, *Sci. Rep.*, 2016, vol. 6, pp. 30139.

12. Chen, Z.C., Yamaji, N., Fujii-Kashino, M., et al., A cation-chloride cotransporter gene is required for cell elongation and osmoregulation in rice, *Plant Physiol.*, 2016, vol. 171, pp. 494–507.

13. Margheri, L., Laschi, C., and Mazzolai, B., Soft manipulator inspired by the octopus: I. From biological functions to artificial requirements, *Bioinspir. Biomim.*, 2012, vol. 7, no. 2, pp. 1–12.

14. Wirekoh, J. and Park, Y.L., Design of flat pneumatic artificial muscles, *Smart Mater. Struct.*, 2017, vol. 26, no. 3, pp. 035009.

15. Rodrigue, H., Wang, W., Han, M.W., et al., An overview of shape memory alloy-coupled actuators and robots, *Soft Robot*, 2017, vol. 4, pp. 3–15.

16. Miriyev, A., Stack, K., and Lipson, H., Soft material for soft actuators, *Nat. Commun.*, 2017, vol. 8, no. 1, pp. 596.

17. Mirvakili, S.M., Sim, D., Hunter, I.W., et al., Actuation of untethered pneumatic artificial muscles and soft robots using magnetically induced liquid-to-gas phase transitions, *Sci. Robot.*, 2020, vol. 5, no. 41, pp. eaaz4239.

18. Neppalli, S., Jones, B., McMahan, W., et al., OctArm - A soft robotic manipulator, *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 2569–2569.

19. Hao, J., Wang, Z., Jin, Y., Chen, X., Peijin, L., Gan, Y., and Chen, X., Hierarchical control of soft manipulators towards unstructured interactions, *Int. J. Robot. Res.*, 2021, vol. 40, pp. 027836492097936.

20. Escande, C., Chettibi, T., Merzouki, R., Coelen, V., and Pathak, P.M., Kinematic calibration of a multisection bionic manipulator, *IEEE/ASME Trans. Mechatronics*, 2015, vol. 20, no. 2, pp. 663–674.

21. Marchese, A.D. and Rus, D., Design, kinematics, and control of a soft spatial fluidic elastomer manipulator, *Int. J. Robot. Res.*, 2015, vol. 35, no. 7, pp. 840–869.

22. Yan, J., Zhang, X., Xu, B., and Zhao, J., A new spiral-type inflatable pure torsional soft actuator, *Soft Robot.*, 2018, vol. 5, no. 5, pp. 527–540.

23. Thuruthel, T.G., Shih, B., Laschi, C., and Tolley, M.T., Soft robot perception using embedded soft sensors and recurrent neural networks, *Sci. Robot.*, 2019, vol. 4, no. 26, pp. eaav1488.

24. Greer, J.D., Blumenschein, L.H., Alterovitz, R., Hawkes, E.W., and Okamura, A.M., Robust navigation of a soft growing robot by exploiting contact with the environment, *Int. J. Robot. Res.*, 2020, vol. 39, no. 14, pp. 1724–1738.

25. Gong, Z., Fang, X., Chen, X., Cheng, J., Xie, Z., Liu, J., and Wen, L., A soft manipulator for efficient delicate grasping in shallow water: Modeling, control, and real-world experiments, *Int. J. Robot. Res.*, 2021, vol. 40, no. 1, pp. 449–469.

26. Kurumaya, S., Phillips, B.T., Becker, K.P., Rosen, M.H., Gruber, D.F., Galloway, K.C., Suzumori, K., and Wood, R.J., A modular soft robotic wrist for underwater manipulation, *Soft Robot.*, 2018, vol. 5, pp. 399–409.

27. Kaufmann, J., Bhovad, P., and Li, S., Harnessing the multistability of kresling origami for reconfigurable articulation in soft manipulators, *Soft Robot.*, 2022, vol. 9, no. 2, pp. 212–223.

28. Schubert, B.E. and Floreano, D., Variable stiffness material based on rigid low-melting-point-alloy microstructures embedded in soft poly(dimethylsiloxane) (PDMS), *RSC Adv.*, 2013, vol. 3, no. 46, pp. 24671–24679.

29. Brown, E., Rodenberg, N., Amend, J., et al., Universal robotic gripper based on the jamming of granular material, *Proc. Natl. Acad. Sci.*, 2010, vol. 107, no. 44, pp. 18809–18814.

30. Kim, Y.J., Cheng, S., Kim, S., et al., A novel layer jamming mechanism with tunable stiffness capability for minimally invasive surgery, *IEEE Trans. Robot.*, 2013, vol. 29, no. 4, pp. 1031–1042.

31. Hao, J., Wang, Z., Jin, Y., Chen, X., Peijin, L., Gan, Y., and Chen, X., Hierarchical control of soft manipulators towards unstructured interactions, *Int. J. Robot. Res.*, 2021, vol. 40, pp. 027836492097936.

32. Bratchikov, S.A., Abramova, E.A., and Fedosov, Yu.V., Solution of the inverse kinematics problem of the manipulator, *Bull. Tomsk State Univ., Manag., Comput. Inform.*, 2021, vol. 56, pp. 4–11. https://doi.org/10.17223/19988605/56/1

33. Molotkov, A.V. and Chelnokov, Yu.N., Solution of the inverse problem of kinematics of the robot-manipulator "Puma" using the biquaternion theory of kinematic control, *Math. Mech.*, 2002, no. 4, pp. 204–206.

34. Antsiporovich, P.P., Akulich, V.K., and Dubovskaya, E.M., Solution of the inverse problem of manipulator kinematics, *Theor. Appl. Mech., Int. Sci. Tech. Collect.*, 2017, no. 32, pp. 306–309.

35. Svinin, M.M., and Uchiyama, M., A new compensation scheme for the inverse kinematics tasks of flexible robot arms, *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, vol. 1, pp. 315–320. https://doi.org/10.1109/ROBOT.1994.351276

36. Lamb, M., Lee, S., Billing, E., Högberg, D., and Yang, J., Forward and Backwards Reaching Inverse Kinematics (FABRIK) solver for DHM: A pilot study, *Proc. 7th Int. Digit. Hum. Model. Symp. (DHM 2022) Iowa Virtual Hum. Summit*, 2022, pp. dhm.31772. https://doi.org/10.17077/dhm.31772

37. Kamilianov, A.R., Planning of motion trajectories of a multi-link manipulator in a complex three-dimensional workspace based on evolutionary methods, Dissertation for the degree of candidate of technical sciences, Ufa, 2007.

38. Plotnikova, N.V., Control of manipulation robots based on fuzzy logic, *Sci. SUSU: Mater. 62nd Sci. Conf.*, 2010, vol. 2, pp. 170–174.

39. Rahim, F.A., Neuro-fuzzy structure for planning the movement of a robot manipulator online in an unknown dynamic environment, *Izv. Vyssh. Uchebn. Zaved., Sev.-Kavk. Reg., Tekh. Nauki*, 2008, no. 6, pp. 41–49.

40. Song, T.L., Lu, Y.P., and Li, Z.Y., Structural design and research of the bionic snake-like robot, *Adv. Mater. Res.*, 2012, vol. 538, pp. 3034–3037.

41. Oluwade, B., Uwadia, C., and Ayeni, J., Asymptotic time complexity of an algorithm for finding the error pattern of a uniform digital code, *J. Sci. Res. Dev.*, 2001, vol. 6, pp. 127–134.

This paper was recommended for publication by R.A. Munasypov, a member of the Editorial Board