Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

A Generic Continuous Multi-Joint Spinal Robotic System for Agile and Accurate Behaviors with GNN-MPC method

Ying Wu[1†], Zida Zhou[1†], Renming Liu[1], Lanxiang Zheng[1] and Hui Cheng[1*]
[†]Equal contribution. Video: https://youtu.be/DDwp12vfYg4
[*]Corresponding Author: chengh9@mail.sysu.edu.cn
[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China

***Abstract*—The biomimetic research of vertebrates is challenging in both mechanism design and control methods. Motivated by natural acrobatics exhibited by cats and humans, this paper presents a generic multi-joint continuous spinal system and a learning-based algorithm for agile and accurate control. The spinal system combines flexibility with a high load-bearing capacity, rendering it suitable for various types of bionic robots. It features a chain-like structure formed by multiple pairs of spherical gear joints, which endow it with the ability to bend in all directions. Then, to realize dynamic and precious control, a universal control framework integrating online and offline learning is proposed. In this framework, Graph Neural Networks are employed to learn the dynamic model parameters of the spine offline, while the parameterized Model Predictive Control (GNN-MPC) can update the dynamic constraints online and select the optimal control strategy. In the aerial flipping task of the spinal column, a dynamic constraint analysis of the angular momentum of the spinal structure is conducted to derive the most efficient flipping strategy. It allows the spinal structure to execute flips in the air without relying on external forces or mechanical structures. Quantitative analyses of high-load applications on the spine reveal that the spinal column can maintain strength, precision and flexibility simultaneously. A series of aerial flipping experiments prove the designed spine's scalability, flexibility and high load capacity. With GNN-MPC, the spine system can realistically mimic biological spine behavior, validating the algorithm's effectiveness and robustness.**

I. Introduction

The vertebral column of biological organisms is essential for complex movements like flexion, extension, lateral bending, and flipping [49][18]. These movements help organisms quickly respond to environmental changes and change directions, crucial for prey capture and predator evasion with agility. Flexible spinal structures allow quadrupedal and humanoid robots to change postures without external force. They have wide-ranging applications in search and rescue [28][17], film special effects [8], military reconnaissance [11], and agriculture [57][34][50]. In space exploration, robots capable of flipping offer greater flexibility and efficiency. They can flip and adjust postures in microgravity for maintenance, repair, and construction inside and outside space stations.

Despite the pivotal role that the spine plays in biological systems, current quadrupedal [15][47] and humanoid robots [22][9] often overlook the spine or reduce the complexity of the spinal mechanism, thereby sacrificing the inherent flexibility of biological systems. This limitation primarily stems from the mechanical challenge of balancing high degrees of freedom (DoF) flexibility with structural strength, coupled with the lack of control algorithms capable of coordinating multi-DoF dynamic responses while adapting to diverse task requirements, ultimately restricting spine's application in quadrupedal and humanoid robots.

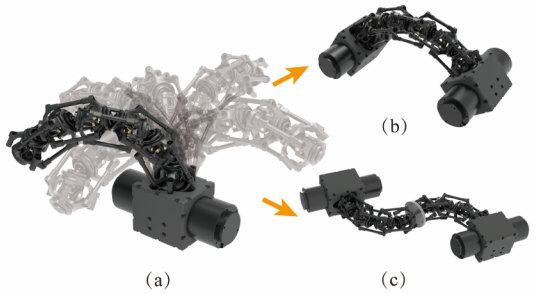


Fig. 1: The 2-DoF Spherical Gear-Based Synchronous Multi-joint Mechanical Spine. (a) Basic mechanism. (b) Install two driving modules to increase the strength. (c) Assemble two mechanisms together for more DoF.

Firstly, in mechanical design, flexible joints with flexibility feature relatively low load-bearing capacity and precision [42][58]. It is difficult to extend their application to spinal robots like quadruped robots. Rigid structures, despite having high load-bearing capacity and precision, require the addition of transmission mechanisms and driving motors to increase their DoF [33][55]. This adds extra weight to the system. Moreover, there is a lack of a spinal structure that can provide both flexibility and strength.

Most studies use predefined models and optimal control algorithms for spinal robots. They show the spine can boost quadruped robot walking, like improving energy efficiency and increasing stride length [56][33]. [7] validates the benefits of the spine's lateral flexion, such as stride extension, better stability, and a smaller turning radius. However, these studies mainly focus on slow-speed tasks and don't explore the spine's dynamic performance for robot flexibility and agility. For example, the spine could help robots do aerial flips or run faster. Thus, it remains challenging to develop control algorithms that fully exploit the dynamic capabilities of spinal joints while ensuring their adaptability across multiple tasks.

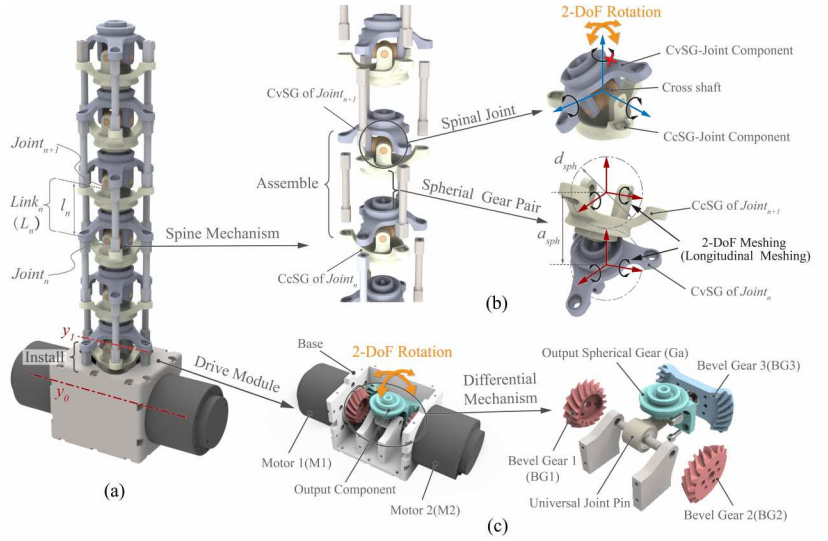


Fig. 2: Mechanical design of the SGb-SMMS. (a) Structure of the SGb-SMMS. (b) Elements and assembly methods of the spine mechanism. (c) Mechanism design of the drive module.

To address the challenge of integrating spinal functionality into quadrupedal and humanoid robots, this paper proposes a novel bionic robotic spine architecture with an associated whole-body dynamic control framework. As illustrated in Fig. 1 and Fig. 2, we develop a 2-DoF Spherical Gear-Based Synchronous Multijoint Mechanical Spine (SGb-SMMS) comprising two subsystems: A mechanical structure using serially connected 2-DoF spherical gear joints enabling omnidirectional bending, and a drive module centered on a bevel gear differential mechanism that dynamically allocates torque between bilateral motors to achieve continuous full-range actuation. The proposed Graph Neural Network-enhanced Model Predictive Control (GNN-MPC) integrates task-specific dynamic constraints with GNN-learned dynamic parameter mappings. We validate the system's efficacy through basic and advanced locomotion tests under diverse conditions. The main contributions of this paper are summarized below:

1) The SGb-SMMS is the first multi-joint spinal architecture integrating spherical gears. By leveraging the multi-directional meshing characteristics of spherical gear pairs, it achieves continuous 3D omnidirectional bending using only two motors. This design synergizes structural strength with flexibility while maintaining control simplicity, and its modular architecture enables seamless scalability.
2) The GNN-MPC employs offline learning with minimal training data to optimize dynamic constraint parameters, combined with online adaptation that updates system parameters based on real-time joint states. This hybrid offline-online strategy enables task adaptability across diverse operational scenarios while fully exploiting the spine's dynamic capabilities for agile maneuvers.
3) We conducted extensive experiments to validate the system's effectiveness, including basic experiments and advanced experiments. The basic experiments verified the structural precision, load-bearing capacity, flexibility and expandability. The advanced experiments analyze the dynamic constraints of feline aerial flipping and achieved multi-posture aerial flipping experiments using the SGb-SMMS, demonstrating the structure's flexibility, scalability, and the algorithm's effectiveness and robustness.

II. RELATIVES WORKS

A. Mechanical design

Biomorphic robots equipped with spine have emerged as a focal point of research within the robotics domain. Such structures enhance robotic agility and stability, enabling the execution of complex maneuvers. The bionic spine in robots is categorized into two distinct types: flexible and rigid. The flexible spine, composed of multiple segments or joints, allows for relative motion between segments, facilitating bending and twisting. These segments are typically actuated by pneumatic and cable systems to mimic the flexibility of biological spines. Cable-driven flexible spines aim to closely replicate the flexibility of vertebral joints, but they are prone to deformation and have limited load-bearing capacity [42] [39] [23]. [58] presents a quadruped soft robot for climbing parallel rods. Its pneumatic-driven spine, via an extendable actuator, offers forward propulsion for effective movement in special environments. Pneumatic systems provide high load-bearing but sacrifice control precision and are greatly affected by environment. The design of flexible spines permits more natural bending and twisting movements in robots, but the challenge lies in maintaining sufficient strength and stability while achieving precise motion control.

Contrasting with the flexible spine is the rigid spine, which is more structurally fixed and robust, to offer significant support and load-bearing capabilities. Rigid spines utilize non-articulating structures and high-strength materials, thereby enhancing their rigidity and durability [48][6]. Despite their limited DoF, these spines excel in stability and force transmission. Serial multi-jointed mechanisms [33] and parallel

mechanisms [55] have been explored to increase the DoF of robotic spines. In particular, the multi-joint spine connected in series has also been studied to mimic the falling cat motion [35]. However, these approaches introduce greater structural complexity and additional weight from extra actuators, potentially compromising the overall flexibility and agility of the robotic system. [2] presents an active ball joint mechanism that combines high rigidity and flexibility through spherical gear interactions, achieving precise positioning and high-torque transmission without sensors. To date, this structure has not been implemented in spinal applications.

B. Control algorithm

In multi-joint robotic research, conventional model-based control methods typically employ predefined models such as the spring-loaded inverted pendulum (SLIP) [12, 14] or preconfigured system states [24]. These approaches enhance computational tractability while preserving adaptability. However, discrepancies between predefined models and real-world dynamics during motion may lead to system instability. For multi-joint systems, accurate whole-body dynamics are generally difficult to obtain, and real-time updates of dynamic parameters are further constrained by computational resources. Consequently, there is growing interest in learning-based approaches that either learn precise system dynamics [45] or integrate model-free methods to directly map robot behaviors to control policies [16].

Current learning-based control methods for multi-joint robots are divided into main approaches: one uses extensive simulation training in virtual environments [5][3][27][44], and the other captures real organism movements for imitation and learning [16][53][25]. While these methods improve robot stability and flexibility, they require extensive training data and time and are often limited to fixed robot configurations. To overcome these limitations, [1][10][4] uses machine learning to learn unknown parameters of dynamics model from few data. With a small amount of flight data, a deep neural network (DNN) can be trained to represent aerodynamic effects under different wind conditions for quadrotors [37]. These methods collect prior information of the system, do not need large datasets and can be adapted to various robotic systems, potentially simplifying dynamic models.

Despite the potentials of active spines to enhance locomotion performance in legged robots, the complexity of spinal models leads to many quadrupedal robots to avoid spinal joints to minimize control difficulties. Common controllers for legged robots with active spines are typically limited to motion in the sagittal plane [12][14]. Recently, compliant spines with lateral flexion capabilities have gained more attention, improving legged robots' turning and walking speed [33][21][7][30]. [33] and [7] design a bioinspired mouse robot, NeRmo, which utilizes a compliant and flexible spine to improve locomotion agility, static stability, walking speed, and maneuverability. However, these studies use open-loop control and trajectory optimization based on simplified models, with spine trajectory pre-defined for specific locomotion tasks. To auto-optimize the spinal trajectory, [26] employs Q-learning in a biomimetic quadruped robot, enhancing spinal dynamic locomotion. To the best of the authors' knowledge, no existing approaches of legged robots have utilized the dynamics of the spine to achieve axial twisting in the sagittal plane.

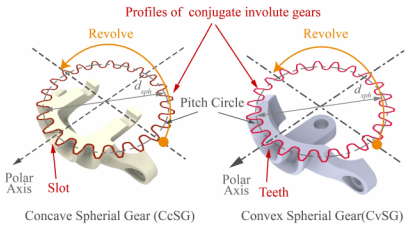


Fig. 3: Involute ring-tooth spherical gear pair.

III. MECHANISM AND KINEMATIC ANALYSIS

The natural animal spine is a continuous, multi-vertebrae structure that bends flexibly through coordinated interactions. Inspired by the natural animal spine, we designed the SGb-SMMS with serially connected spinal joints that diven by a drive module at one end of the mechanical spine allows for simultaneous joint movement. This section presents the mechanical design and kinematic analysis of the SGb-SMMS, showing how it mimics biological spine movement via mechanical linkages.

A. Mechanism design

The structure of the SGb-SMMS is shown in Fig. 2, which is mainly composed of two parts: the spine mechanism and the drive module.

The spine mechanism is designed based on spherical gear pairs. Spherical gear is a special type of gears whose teeth are distributed on the surface of a sphere and are used to achieve the transmission of multi-DoF motions in space [59][51][52]. Due to the meshing of the tooth surface structure, it has high torque transmission and reliable positioning. Thus, spherical gear have great potential and advantages in the application of multi-DoF robot joints [40][32][20]. The spherical gear pair applied in the spine mechanism is the involute ring-tooth spherical gear pair [13]. There are abundant studies on the kinematics [38], strength analysis [43][54] and manufacturing processes [46] of this type of spherical gear, which proves that it has certain advantages over other types of spherical gears. For example, it has relatively good load-bearing capacity and transmission accuracy, is free from errors in the transmission principle, is easier to manufacture, and most importantly, meets the requirements of the 2-DoF transmission.

As shown in Fig. 3, the involute ring-tooth spherical gear pair is formed by rotating the tooth profiles of a pair of conjugate involute spur gears around their polar axes. Since the meshing of gears requires the teeth and tooth slots of two gears to correspond to each other, one spherical gear's polar axis is the symmetry axis of a tooth, and the other's polar axis is the symmetry axis of a tooth slot. According to the central structures, they can be divided into convex spherical gear (CvSG) and concave spherical gear (CcSG).

As shown in Fig. 2, the spherical gears are the main structure of the spinal joint components. Cut one end of the spherical gear in a circular shape along the polar axis and distribute several support brackets evenly around it. Then, arrange hinged supports near the center of the sphere, and a joint component is formed. According to the structure of the spherical gear, the joint components can be classified into CvSG-joint component and CcSG-joint component. A spinal joint is assembled by connecting a CcSG-joint component and a CvSG-joint component on a cross shaft. The two joint components can rotate in 2-DoF along the center of the joint relative to each other. Meanwhile, these two types of joint components can form spherical gear pairs. When forming a spherical gear pair, the two spherical gears are installed opposite to each other along the polar axes, and their pitch sphere are tangent, and the distance a_{sph} between their spherical centers is the sum of the pitch circle radius of the two spheres. The spherical gear pair demands the two gears have the same modulus, and the tooth ratio of the spherical gear pairs in the spine mechanism is $1:1$. Thus the two gears have the same pitch circle diameter d_{sph}:

$$a_{\text{sph}} = d_{\text{sph}}. \tag{1}$$

The two meshing sphercial gears can mesh along any longitude direction, doing 2-DoF pure rolling. At the contact point, the speeds of the two spherical gears are equal and opposite. As a result, the two joint components can be bent in any direction relative to each other.

When assembling the spine mechanism, the polar axes of all spinal joints are in a coaxial state, and the spinal joints are aligned in the same direction. Using rods to connect the two adjacent spinal joints by assembling the components that do not participate in the meshing between these two joints. Since the joints' orientation are the same, there is a CvSG and a CcSG between every two joints. Set the distance l_n between the joints to be equal to the center distance of the spherical gear pair:

$$l_n = a_{\text{sph}}. \tag{2}$$

So that the CvSG and the CcSG between the joints mesh with each other to form a spherical gear pair. In this way, the spine mechanism is assembled.

Fig. 2(c) shows the mechanism design of the drive module. It mainly uses a bevel gear differential mechanism which has advantages in the application of multi-DoF robotic joints as it can achieve multi-DoF motion and flexibly distribute power [29][36]. In the driving module, two motors (M1,M2) on both sides make the output component rotates through the differential mechanism. In the differential mechanism, two identical bevel gears BG1 and BG2 are distributed on both sides and respectively mounted on M1 and M2. At the center of the mechanism is a cross shaft, one of its axes is hinged to the base and is coaxial with BG1 and BG2, while the other axis is hinged to the actuator. The actuator is composed of a spherical gear component (G_a) and a bevel gear (BG3) fixed together. In the straightened state of the spine, the axis of the spherical gear is along the z-axis, and the axis of BG3 is along the x-axis. BG1 and BG2 are respectively meshed on both sides of SG3, and through rotational coupling, drive the output component to rotate omnidirectionally.

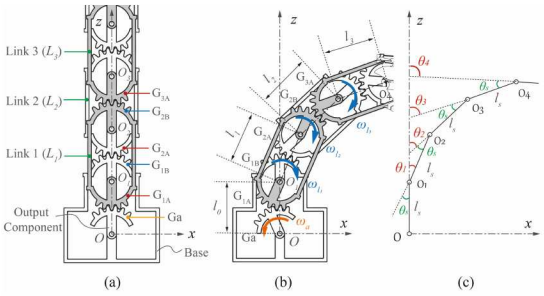


Fig. 4: Sectional view of the spinal mechanism. (a) Extended state. (b) Bending motion. (c) Geometric analysis.

To install the spine mechanism on the drive module, engage the output component with the spherical gear at the end of the spine mechanism and fix another joint component of the end spine joint to the base of the drive module as shown in Fig. 2(a). Then a basic SGb-SMMS is assembled. Additionally, depending on actual needs, we can install drive modules at both ends of the spine mechanism to obtain greater driving force as Fig. 1(b), or assemble multiple SGb-SMMS together to achieve more DoFs as Fig. 1(c).

B. Kinematic analysis

This subsection introduces the mechanical principle of the spine mechanism, and demonstrate how multiple joints can synchronously carry out a bending movement with 2-DoF through mechanical transmission, so that it can achieve the movement state of the biological spinal structure.

To explain the movement principle of the spine more clearly, we first conduct the analysis from 1-DoF. Taking the movement of the spine in the x-axis direction as an example, the plan view of the mechanism movement is shown in Fig. 4. First, let's explain the meaning of the symbols in the Fig. 4: G_a represents the spherical gear of the output component, and G_{1A}, G_{1B}, G_{2A}... represent the spherial gears on the spinal joints. In the subscript symbol of G_{1A}, '1' indicates the first joint, 'A' represents the spherical gear whose polar axis points to the G_a, and 'B' represents another spherical gear on the same joint. According to the mechanism design, the link between two joints L_n is composed of the spherical gears facing outwards of two adjacent joints. For example, L_1 is composed of G_{1A} and G_{1B}. Since they form a whole, their rotation speeds are the same during movement.

As it is a planar motion, the spherical gears can be analyzed as involute spur gears. In Fig. 4, when G_a rotates counter-clockwise at ω_a, since the gear ratio of all gears is $1:1$, G_{1A} rotates in the opposite direction at the same speed, driving L_1 to rotate with angular velocity ω_{l1}:

$$\omega_{l1} = \omega_a. \tag{3}$$

Denote the rotation speed of the spherial gears on the spinal joint G_{1B} and G_{2A} as ω_{1B} and ω_{2A}. G_{1B} is fixed on the base and stationary, so that $\omega_{1B} = 0$. G_{2A} is hinged on O_2, and O_2

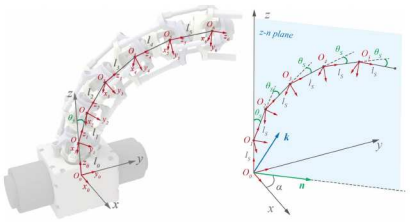


Fig. 5: 2-DoF movement of the spine mechanism.

rotates with L_1. Therefore, G_{1B}, L_1 and G_{2A} form a planetary gear mechanism. According to the analysis of the motion of the epicyclic gear train, the reduction ratio i_{BA}^{L1} of 'B' to 'A' of L_1 frame can be obtained:

$$i_{BA}^{L1} = \frac{\omega_{1B} - \omega_{l1}}{\omega_{2A} - \omega_{l1}}. \tag{4}$$

Since the teeth number of the gears are the same, the transmission ratios i_{BA}^{Ln} of all gear pairs are -1. Based on the given conditions, we can calculate the rotation speed of L_2: $\omega_{L_2} = \omega_{2A} = 2\omega_a$ and the direction of rotation is the same as that of L_1. Thus, the rotational speed of L_2 relative to L_1 is ω_a. By analyzing the subsequent joints in the same way, the rotational speed ω_{Ln} of each link and the relative speed between two adjacent links can be obtained:

$$\begin{aligned} \omega_{Ln} &= n\omega_a \\ \omega_{Ln} - \omega_{L(n-1)} &= \omega_a. \end{aligned} \tag{5}$$

The geometric relationship of the spine mechanism during movement can be obtained as shown in Fig. 4(c). When G_a rotates by θ_s, all joints will synchronously rotate by θ_s relative to the previous joint, and the rotation angles θ_n of each joint are:

$$\begin{aligned} \theta_n &= n\theta_s \\ \theta_n - \theta_{(n-1)} &= \theta_s. \end{aligned} \tag{6}$$

The universal joint enables two-DoF rotation of the spherical joint, allowing the spherical gear pair to mesh in all longitude directions. This makes the planar mechanism's motion possible on planes through the z-axis, enabling the spine to bend in any $x-y$ plane direction (Fig. 5). The spine's bending posture is determined by θ_s (joint rotation angle, controlling bending amplitude) and α (deflection angle, determining bending direction). Due to reduced contact area and meshing reliability when involute ring-tooth spherical gears mesh, θ_s is limited to 30°, and $\alpha \in (0°, 360°)$.

All the joints and links of the spine mechanism are located in the $z-n$ plane and follows the motion principle of the planar mechanism described in Fig. 4. Therefore, each joint rotates by θ_s in the same direction, and the rotation axis of the joint is the vector $\boldsymbol{k}$ perpendicular to the $z-n$ plane.

$$\boldsymbol{k} = \boldsymbol{z} \times \boldsymbol{\alpha} \tag{7}$$

$$\alpha = \arccos\left(\frac{\boldsymbol{n} \cdot \boldsymbol{x}}{|\boldsymbol{n}||\boldsymbol{x}|}\right) \tag{8}$$

In Fig. 6, the direction of each link of the spine is the $\boldsymbol{z_n}$ direction of respective joint coordinate system. According to

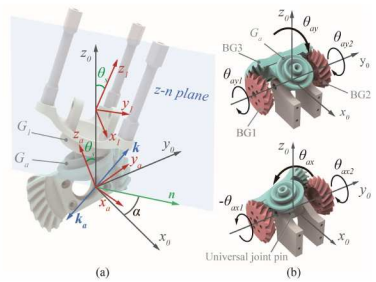


Fig. 6: Drive device motion analysis. (a) The motion of the spherical gear pair between the spine mechanism and the drive module. (b) Kinetic analysis of bevel gear differential mechanism.

Eq. (5), the relationship of the rotation angles of each link can be known, and the calculation formula for $\boldsymbol{z_n}$ can be obtained:

$$\boldsymbol{z_n} = \mathrm{Rot}(\boldsymbol{k}, n\theta_s)\boldsymbol{z_0}, \tag{9}$$

where the rotation matrix $\mathrm{Rot}(\boldsymbol{k}, n\theta_s)$ represents a rotation of angle $n\theta_s$ around the vector $\boldsymbol{k}$, and can be solved by the Rodriguez formula.

Due to the existence of gimbal lock in the universal joint used in the spinal joint, the rotational change of the joint coordinates needs to be determined by the installation direction of the cross shaft. When installed as shown in the Fig. 2, the hinge axis of the joint component fixed to the drive module is parallel to the y-axis, then the x direction movement of l_1 is carried out along the coordinates of the universal joint pin. The angles of rotation of the spinal joints along the x-axis and y-axis are denoted as θ_{sx} and θ_{sy} respectively. At the same time, since the installation directions of each spinal joint are the same, it can be obtained that the rotation matrices R_s of each joint are the same, and can obtain the rotation matrix as follow:

$$\boldsymbol{R_s} = \mathrm{Rot}(\boldsymbol{y}, \theta_{sy})\mathrm{Rot}(\boldsymbol{x}, \theta_{sx}) \tag{10}$$

Substituting Eq. (9) and Eq. (10) into L_1 can solve θ_{sx} and θ_{sy}, thereby the rotation matrices of all joints can be obtained. It is known that the lengths l_s of all links are the same. Therefore, the transformation matrix ${}^{n-1}\boldsymbol{T}_n$ between every adjacent joints can be obtained:

$${}^{n-1}\boldsymbol{T}_n = \begin{pmatrix} & \boldsymbol{R_s} & & 0 \\ & & & 0 \\ & & & l_s \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{11}$$

Based on the transformation matrix ${}^{n-1}\boldsymbol{T}_n$, the posture relationships between spine links can be obtained. Matrix multiplication determines the posture of any joint and points on it relative to the driving device.

When the drive module drives the spine mechanism, the spherical gear pair formed between the driving device and the spine mechanism is shown in Fig. 6(a). The transmission between the two spherical gears is relative to pure rolling. On the cross section of the spherical gear movement direction ($z-n$ plane), G_a rotates in one direction, then G_1 rotates in the opposite direction at the same angle. In Fig. 6, the spherical

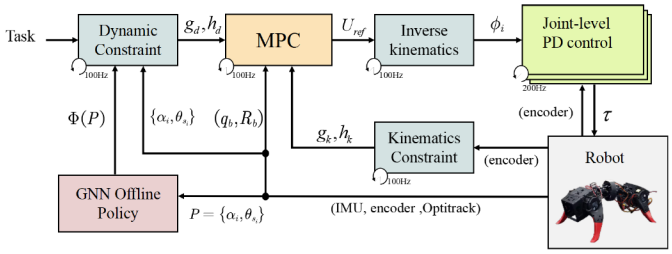


Fig. 7: The control frame of the GNN-MPC.

gear G_1 rotates θ_s along $\boldsymbol{k}$, while G_a rotates along $\boldsymbol{k_a}$, which is opposite to $\boldsymbol{k}$. It can be obtained that the polar axis vector of G_a as follows:

$$\begin{aligned} \boldsymbol{k_a} &= -\boldsymbol{k} \\ \boldsymbol{z_a} &= \text{Rot}(\boldsymbol{k_a}, n\theta_s)\boldsymbol{z_0}. \end{aligned} \tag{12}$$

The movement of G_a is controlled by the bevel gear differential mechanism. As shown in Fig. 6, when the BG1, BG2 rotate by the same angle θ_{ay1}, θ_{ay2} in the same direction, they drive the universal joint pin and BG3 to rotate by θ_{ay} along $\boldsymbol{y}_0$.

$$\theta_{ay} = \theta_{ay1} = \theta_{ay2}. \tag{13}$$

When the BG1 and the BG2 rotate in the opposite direction by the same angles θ_{ax1} and θ_{ax2} respectively, it causes the BG3 to rotate by θ_{ax} along the axis of the universal joint pin. In addition, since the number of teeth of the BG1, BG2 is different from that of the BG3, there is a reduction ratio i_b in the transmission.

$$i_b = \frac{z_{b3}}{z_{b1}} = \frac{z_{b3}}{z_{b2}} \tag{14}$$

$$\theta_{ax} = -\frac{\theta_{ax1}}{i_b} = \frac{\theta_{ax2}}{i_b}. \tag{15}$$

The two kinds of motion can occur simultaneously, enabling G_a to perform omnidirectional rotational motion. The transformation matrix ${}^0T_a$ of the G_a is:

$${}^0\boldsymbol{T}_a = \text{Rot}(\boldsymbol{y}, \theta_{ay})\text{Rot}(\boldsymbol{x}, \theta_{ax}). \tag{16}$$

For the rotation angles θ_{b1} and θ_{b2} of the two bevel gears BG1 and BG2, when the driver rotates, each of them is composed of the two basic rotations when G_a rotates omnidirectionally.

$$\begin{aligned} \theta_{b1} &= \theta_{ax1} + \theta_{ay1} \\ \theta_{b2} &= \theta_{ax2} + \theta_{ay2}. \end{aligned} \tag{17}$$

The rotation angles of BG1 and BG2 are the output angles of motors M1 and M2. The corresponding relationships of the movement of the driven components driven by the motors can be obtained.

$$\begin{pmatrix} \theta_{b1} \\ \theta_{b2} \end{pmatrix} = \begin{pmatrix} -i_b & 1 \\ i_b & 1 \end{pmatrix} \begin{pmatrix} \theta_{ax} \\ \theta_{ay} \end{pmatrix}. \tag{18}$$

Therefore, by substituting the specific values θ_s and α into Eq. (10)-(18), the required motor output angles θ_{b1}, θ_{b2} can be obtained. So that the drive module can perform accurate, reliable and real-time control of the spine mechanism as shown in Fig. 2.

IV. LEARNING BASED MPC CONTROLLER

While an active spine boosts the agility, stability, and flexibility of multi-joint robots, the extra DoF of spine adds significantly complicate the dynamics model. In addition, the external forces acting on the robot vary in different situations, and the role of the spine also changes, which results in different dynamic models of the robot. For example, in the air, the robot can use its spine to flip, while on the ground, it can achieve faster running speeds through spine extension and contraction. As depicted in Fig. 7, to fully utilize the SGb-SMMS's capabilities within limited computational power, this section details a GNN-MPC method that combines offline learning and online update for the spinal system consisting of j SGb-SMMS in series.

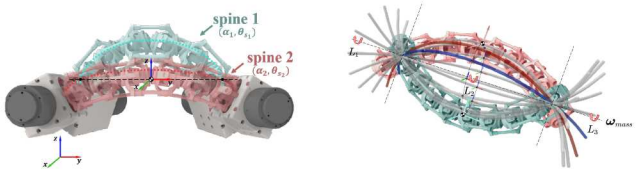
A. Adaptive MPC

As shown in Fig. 5, we use the bending angle θ_s of the spine and the rotation angle α to describe the rotational state of the SGb-SMMS. The spinal robot system state is given by $q_b = \{\boldsymbol{R}_b, \boldsymbol{P}_b\}^T$, θ_{si} and α_i represent the state of the i-th SGb-SMMS, and the control outputs are chosen as $\boldsymbol{U}_{\text{ref}} = \{\dot{\alpha}_i, \dot{\theta}_{si}\}^T$ with $(j \geq i \geq 1)$, where the rotation matrix $\boldsymbol{R}$ represents the posture of the robot's center of mass (CoM) and $\boldsymbol{P}$ is the position of the robot's CoM. The optimization problem takes the desired state $\boldsymbol{q}_{b,\text{ref}}$ as input and produces the SGb-SMMA trajectories $\boldsymbol{U}_{\text{ref}}$ as output. Therefore, we construct the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{U}_{\text{ref}}}{\arg\min} \quad & \left\| \left(\boldsymbol{P}_b \boldsymbol{P}_{b,\text{ref}}^T \right) \right\|_{\boldsymbol{K}_P}^2 + \left\| \left(\log(\boldsymbol{R}_b \boldsymbol{R}_{b,\text{ref}}^T) \right)^{\nu} \right\|_{\boldsymbol{K}_r}^2 \\ & \boldsymbol{g}_k(q_b, U_{\text{ref}}) \geq 0 \\ & \boldsymbol{g}_d(q_b, U_{\text{ref}}) \geq 0 \\ & \boldsymbol{h}_k(q_b, U_{\text{ref}}) = 0 \\ & \boldsymbol{h}_d(q_b, U_{\text{ref}}) = 0, \end{aligned} \tag{19}$$

where the $\boldsymbol{g}_k$ and $\boldsymbol{h}_k$ represent the kinematic constraints of the spine robot, $\boldsymbol{g}_d$ and $\boldsymbol{h}_d$ represent the dynamic constraints of the spine robot. The spine functions differently in various scenarios, and the corresponding dynamic constraints are also distinct. To enhance computational efficiency, our control system analyzes the dynamic model of the spine in accordance with different scenarios and selects the most efficient dynamic constraints. In order to analyze the dynamic model of the spine in various situations, accurate dynamic parameters are essential. In [41], a Body Transformer architecture based on GNN is proposed for learning robot policies. Its advantage lies in the ability to capture the structural information and relationships of multi-joint robots. Therefore, we adopt a model parameter identification method based on GNN.

In contrast to the dynamic constraints, the kinematic constraints of the same spinal joints remain unaltered. As shown in Fig. 5, the spine can be modeled as a series of connected, compatible revolute joints (n-DoF). Due to the constraints of the spinal joints, the rotation angles of each revolute joint are consistent. Denote the transformation matrix of the i-th SGb-SMMS as ${}^0\boldsymbol{T}_a^i(\alpha_i, \theta_{si})$. Therefore, we transform the kinematic



(a) different bending angles (b) different α

Fig. 8: (a): Different values of θ_s result in different bending angles of the spine, but the center of mass of the spine remains unchanged. (b): With the same value of θ_s, changing only α causes the spine to rotate around a fixed axis at the same bending angle, generating rotational angular momentum.

constraints of the spinal system consisting of j SGb-SMMS in series. The constraints on the relative positions are refined in terms of the relationship between the positions of the spinal end $\boldsymbol{P}_n$ and the state of the spine $\{\alpha_i, \theta_{si}\}, 0 < i \leq j$.

$$\begin{aligned} \boldsymbol{P}_n &= {}^0\boldsymbol{T}_a^1(\alpha_1, \theta_{s1}) \cdots {}^0\boldsymbol{T}_a^j(\alpha_j, \theta_{sj})\boldsymbol{P}_0 \\ \boldsymbol{P}_{\text{lower}} &\leq \boldsymbol{P}_n \leq \boldsymbol{P}_{\text{upper}}, \end{aligned} \tag{20}$$

where $\boldsymbol{P}_0 = [0 \quad 0 \quad l_n \quad 1]^T$. And Eq. (20) represents the limitations of the workspace at the spinal end, and as j increases, the workspace at the spinal end becomes larger. However, the load and accuracy of the spinal system also decrease. By specifying dynamic constraints for different tasks and solving the optimization problem, we can obtain the desired output U_{ref} of the spinal system. Through inverse kinematics calculation, the state corresponding to U_{ref} can be transformed into the desired angle ϕ_i and angular velocity $\dot{\phi}_i$ of each joint motor. The joint trajectory is controlled by a Proportional-Derivative control (PD control):

$$\tau_i = K_p(\phi_{i,d} - \phi_i) + K_d(\dot{\phi}_{i,d} - \dot{\phi}_i), \tag{21}$$

where τ_i is the control torque of the i-th joint, K_p is the proportional gain coefficient, K_d is the derivative gain coefficient, $\phi_{i,d}$ is the desired angle of the i-th joint, and $\dot{\phi}_{i,d}$ is the desired angular velocity of the i-th joint.

B. Learning-based Dynamic Constraints

The dynamic role of the spine in increasing the turning radius and walking speed has been widely studied. Therefore, this paper mainly focuses on the dynamic constraints of the spinal robot's flipping only through the spine. Since the falling cat's mid-air flip relies on the Law of Conservation of Angular Momentum (LOCAM), LOCAM serves as the fundamental principle of the spinal robot's flipping. The LOCAM states that the total angular momentum of a spinal robot remains constant if no external torques act on the system when it falls from the air. Assume that the initial momentum and angular momentum of the robot in the air are both 0. In fact, different θ_s and α can change the bending angle of the spine and also make the spine rotate around a fixed axis. Since the robot is only affected by the vertical downward gravity in the air, the position of the robot's center of mass only moves downward along the z-axis of the world coordinate system. As shown in Fig. 8, assuming that the θ_s of the spine at a certain point in the air is different, resulting in different curvatures of the spine, the center of mass of the spine remains unchanged. As shown in Fig. 8(b), if appropriate θ_s are selected to make the bending angles of the spine in the air the same and only change the rotation angle α of the spine around the fixed axis, the spine rotates around the rotation axis $\boldsymbol{\omega}_{mass}$ passing through the center of mass. As shown in Fig. 8(b), assuming that the spine rotates around the rotation axis passing through the center of mass with different bending angles, then we can divide the spine into three parts. The angular momenta generated by the three parts around $\boldsymbol{\omega}_{mass}$ are L_1, L_2 and L_3 respectively. Since the directions of the angular momenta generated by L_1, L_2 and L_3 are the same:

$$L = L_1 + L_2 + L_3 \neq 0. \tag{22}$$

According to the LOCAM, there must be another angular momentum in the robot system that is equal in magnitude and opposite in direction to L to cancel it out, so that the angular momentum of the entire system remains constant. Since the robot is symmetric about the plane that passes through its center of mass and is perpendicular to $\boldsymbol{\omega}_{mass}$, to nullify the angular momentum L, the robot can only rotate about its own axis to alter its attitude relative to the world coordinate system. Denote the angular momentum generated by the robot's rotation as L_B, then

$$L = L_B. \tag{23}$$

Since the angular momentum is the product of the moment of inertia and angular velocity:

$$\begin{aligned} L &= L_1 + L_2 + L_3 \\ &= J_1\omega_1 + J_2\omega_2 + J_3\omega_3 \\ L_B &= J_B\omega_B. \end{aligned} \tag{24}$$

Because the spine as a whole rotates around $\boldsymbol{\omega}_{mass}$ at the same speed, that is $\omega_1 = \omega_2 = \omega_3$. Since the moment of inertia is related to the mass distribution of the rotating object around the rotation axis, different α and θ_s of the spine correspond to different moments of inertia. That is, denote the rotation angle of the spine around the rotation axis $\boldsymbol{\omega}_{mass}$ as ω_{mass}, i.e.

$$\begin{aligned} J(\alpha, \theta_s)\omega_{mass} &= J_B\omega_B \\ J(\alpha, \theta_s) &= J_1 + J_2 + J_3. \end{aligned} \tag{25}$$

According to the law of conservation of energy, $\omega_{mass} \geq \omega_B$. As shown in the Fig. 8(b), when $J(\alpha, \theta_s) = J_B$, we can get $\omega_{mass} = \omega_B$, that is, $\frac{1}{2}J(\alpha, \theta_s)\omega_{mass}^2 = \frac{1}{2}J_B\omega_B^2$. The kinetic energy generated by the rotation of the spine is completely converted into the kinetic energy of the robot's rotation, which not only has the highest flipping efficiency but also no energy loss. However, if $J(\alpha, \theta_s) \ll J_B$, we can get $\omega_{mass} \gg \omega_B$, $\frac{1}{2}J(\alpha, \theta_s)\omega_{mass}^2 \gg \frac{1}{2}J_B\omega_B^2$, then the kinetic energy generated by the rotation of the spine is almost completely lost, the angular velocity of the spine rotation is large, and the rotation amplitude of the robot itself is also very small. Therefore, the

smaller $J(\alpha, \theta_s)$ means, the lower the efficiency of the robot's rotation and the lower the energy conversion efficiency.

Therefore, we choose Eq. (25) as the dynamic constraint for the robot's flipping in the air. The solver considers the relationship between the moment of inertia J and the spine states α, θ_s, and can solve for the control output with the highest flipping efficiency.

As shown in Fig. 1, due to the expandability of the SGb-SMMS, two SGb-SMMS are connected in series. Then, since the two ends of the series-connected SGb-SMMS can rotate in different directions respectively, the series-connected spine has angular kinetic energy for rotation in different directions. Therefore, when extending this dynamic constraint to a multi-segment series-connected spine, the LOCAM is as follows:

$$\begin{aligned} &J^1(\alpha_i, \theta_{si})\omega^1 + J^2(\alpha_i, \theta_{si})\omega^2 = J_B\omega_B \\ &J(\alpha, l) = J_1 + J_2 + J_3. \end{aligned} \tag{26}$$

The MPC solver considers the relationship between the moment of inertia $J^i(\alpha_i, \theta_{si})$ and the spine joint angles, and can solve for the control output with the highest flipping efficiency.

C. Training of the GNN

Due to the fact that we only consider the flexibility of individual SGb-SMMS connected in series, and the connections between joints are relatively simple, the GNN can be designed as a 3-layer structure.

Based on the dynamic constraints of aerial flip, we make full use of the powerful learning ability of GNN to study the moment of inertia of the spine at different bending angles. We import the URDF model of the spinal model into Gazebo, and set the Gazebo environment to a zero-gravity environment. By setting different initial positions of the spinal joints $\{\alpha, \theta_s\}$ and various rotation speeds ω_{mass}, we get the angular velocity of spinal self-rotation ω_B in Gazebo.

Then the relationship between the moment of inertia as:

$$J_i = \frac{J}{J_B} = \frac{\omega_B}{\omega_{mass}} \tag{27}$$

Thus, we can obtain a series of datasets $D_k = \{P_1, P_2, \cdots P_n\}$, and $D_J = \{J_1, J_2, \cdots J_n\}$ corresponding to the points of this dataset. And the designed 3-layer GNN consists of an MLP layer, a feature aggregation layer, and an MLP layer.

After training, the variation relationship between the single spinal joint J_i and the spinal state $\{\alpha, l\}$ is obtained in Fig. 9. As analyzed theoretically, the moment of inertia $J$of the SGb-SMMS increases with the growth of the bending angle. At the same ω_{mass}, the spinal self-rotation speed ω_B rises, enabling more efficient conversion of the motor-generated angular kinetic energy into spinal self-rotation kinetic energy.

Given this complex situation, it is extremely difficult to determine the inherent relationship J_i between the dynamic parameters and the joint angle through conventional dynamic analysis methods, such as Lagrange methods [31]. The classical methods can only estimate J_i by approximating the spherical gear as an equivalent regular rigid body.

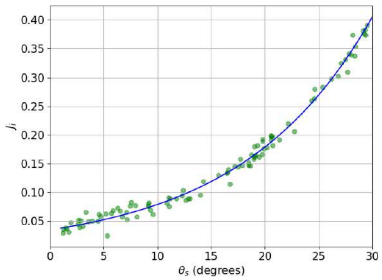


Fig. 9: The relationship between a single spinal joint J_i and the spinal state θ_s.

TABLE I: The MSE of different methods for SGb-SMMS.

methods	equivalent	MLP	GNN
$n_g < 6$	2.0×10^{-2}	$< 1.0 \times 10^{-4}$	$< 1.0 \times 10^{-4}$
$n_g \geq 6$	3.0×10^{-2}	1.2×10^{-3}	$< 1.0 \times 10^{-4}$

We employed the classical equivalent rigid-body method (equivalent), MLP, and GNN networks to predict the J_i for SGb-SMMS with number of spherical gears n_g ranging from 4 to 10. GNN and MLP are designed with nearly identical parameter counts (500 epochs and a learning rate of 0.01), using the Mean Squared Error (MSE: $\frac{1}{n}\sum(y - \hat{y})^2$ where n is batchsize, $y - \hat{y}$ represents the predicted error). In Table I, the network-based methods significantly outperform the equivalent method. MLP performs well when $n_g < 6$ but degrades with more serial gears. GNN shows good generalization and inverse-deduction abilities for over 5 serial gears.

V. Experiment Results

In experiments, we use various robots with the SGb - SMMS to test the spinal system's versatility, dexterity, and robustness. The experiments, divided into basic (precision-load and flexibility) and advanced (aerial flipping and robustness) types, are detailed below:

1) Precision-load experiment: The SGb-SMMS carried weights and rotated about a fixed axis to verify its strength, precision, and flexibility.

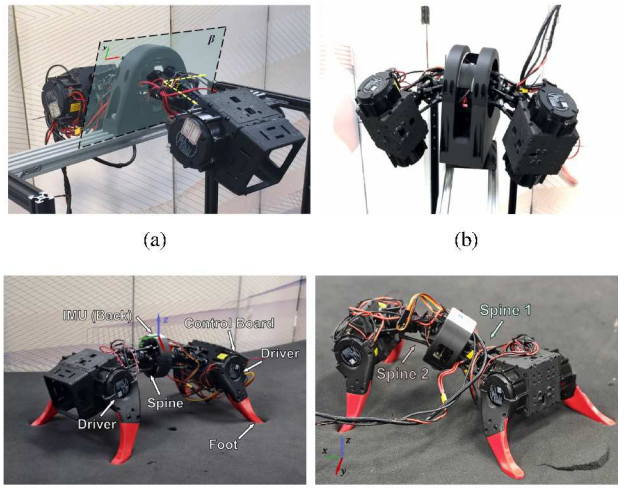
2) Flexibility experiment: In the flexibility experiment, two SGb-SMMS units were spliced to test the DoF of the combined spine rotating around various axes, verifying its structural expandability and enhanced flexibility.

3) Aerial flipping experiment: In this experiment, both a single-SGb-SMMS robot and a double-SGb-SMMS robot were dropped from a height in different initial postures to verify that the spine and algorithm can mimic the flexible structure and behavior of natural spinal organisms, as well as the effectiveness of enhancing flexibility through serial connection.

4) Robustness experiment: In this experiment, we drop the single-SGb-SMMS robot from a height with randomly designed initial angular kinetic energies, then conduct comparative and repeated tests against other methods to validate the robustness and reliability of our proposed algorithm.

A. Experiment setup

In this experiment, four experimental platforms are utilized. Fig. 10(a) represents the platform for the precision-load exper-



(a) (b) (c) (d)

Fig. 10: experimental platform. (a): Spinal strength and precision test platform. (b): Dual-SGb-SMMS connection flexibility test platform. (c): The single-SGb-SMMS robot. (d): The double-SGb-SMMS robot consists spine 1 and spine 2.

iment, which is composed of one segment of SGb-SMMS. It has two degrees of freedom and can rotate around a fixed axis. Fig. 10(b) shows the experimental platform for the flexibility experiment. This platform has DoFs to rotate around multiple axes. Fig. 10(c) shows a single-SGb-SMMS robot, in which the spinal joints can only bend at the same angle. Fig. 10(d) presents a double-SGb-SMMS robot. It has 3-DoF, and the two segments of SGb-SMMS can bend in different directions.

The spine of the robot is equipped with Wit's six-axis IMU to obtain the robot's posture and acceleration accurately. Meanwhile, an Optitrack motion tracking system is employed to gather information such as the position, velocity, and angular velocity of the robot's movement. The robot is controlled by a NUC computer. The driving motor for the spinal joints is a Gyber motor, and the solver for the MPC optimization problem is Acados.

B. Analysis of precision-load experiment

Since the length and load of the SGb-SMMS are inversely proportional to the precision, it is a great challenge to balance the flexibility, precision, and strength of the spinal joints. The platform for the load experiment is shown in Fig. 10(a). When unloaded, the spine bends downward by γ due to the motors' gravitational pull. We then perform circular motion around the bent spine axis; the trajectory's projection on plane β perpendicular to the spinal axis is circular. Finally, we convert the target trajectory to the world coordinate system and, via inverse kinematics, transform it into spinal joint control angles.

Because the spinal structure is rigid, its load-bearing capacity is determined by the driving motors. The spinal joints employ GyberGear motors, which have a continuous torque of $4\ N \cdot m$, a spinal joint length of $0.40\ m$, and a motor self - weight of $0.317\ kg$. Theoretically, the SGb-SMMS load should not exceed $1.3\ kg$. We performed load experiments on the SMMS with loads of $0\ kg$ and $1.2\ kg$.

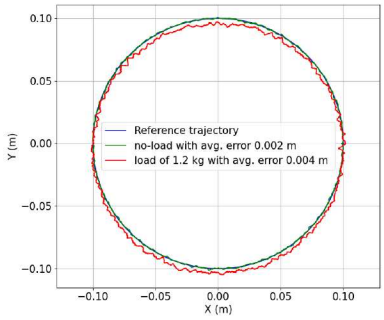


Fig. 11: The trajectory of the projection of the end of SGb-SMMS on the β plane.

It can be seen in Fig. 11 that the SGb-SMMS can flexibly and stably follow the circular trajectory under free-load and a load of $1.2\ kg$. With no load, the SGb-SMMS tracks the target trajectory accurately, with an average error of within $0.002\ m$. When the load is $1.2\ kg$, the spine maintains accurate tracking with an error within $0.004\ m$. Therefore, this experiment proves that our SGb-SMMS structure 1) can flexibly track circular trajectories, 2) has a large load-bearing capacity, and 3) features high precision.

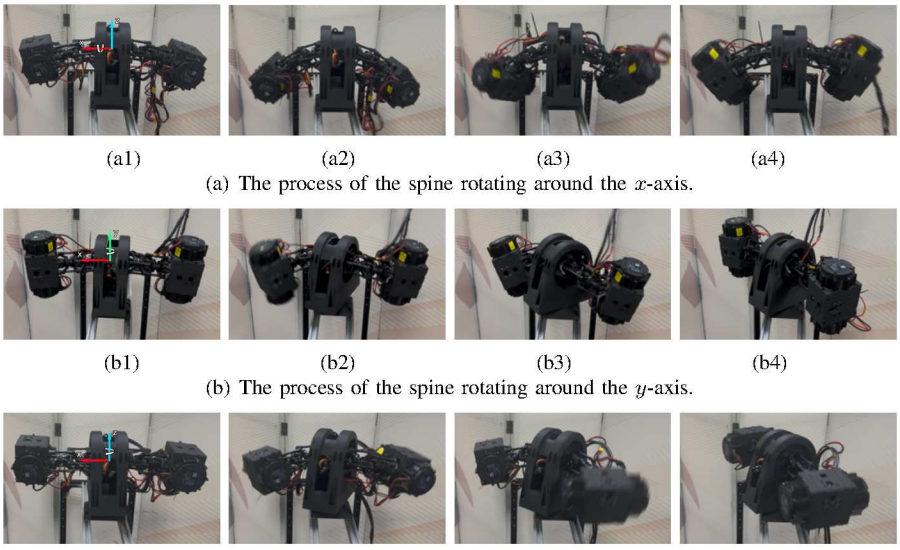
C. Analysis of flexibility experiment

The end of a single-SGb-SMMS can rotate around a fixed axis by the LOCAM. In the tandem double-SGb-SMMS, the two ends can rotate in different directions, giving it more DoFs for multi-axis rotation. To verify the flexibility and expandability of the proposed spine structure, the experiment has three parts: 1) Double-SGb-SMMS rotates around the body's x-axis via joint movement; 2) Around the y-axis; 3) Around the z-axis.

As shown in Fig. 12(a), the double-SGb-SMMS can achieve the self-rotation around the body's x-axis by continuously bending two SGb-SMMS in the same direction by the same angle. In Fig. 12(b), The double-SGb-SMMS can achieve the self-rotation around the body's y-axis by first rotating one SGb-SMMS around the y-axis, then rotating the other SGb-SMMS around the y-axis, and finally restoring both SGb-SMMS to their initial states simultaneously. Similarly, the spinal joints can rotate around the body's z-axis as shown in Fig. 12(c). Therefore, we have verified the expandability and higher flexibility of the SGb-SMMS proposed in this paper. Its modular design can be applied to different spine robots according to the needs of practical applications, enabling more flexible movements.

D. Analysis of aerial flipping experiment

A single SGb-SMMS can flip in one direction, allowing the spinal robot to land legs-down from a high-altitude back-down drop. The double-SGb-SMMS robot has more flexible degrees of freedom, enabling it to flip from back-down to back-up and rotate around the body's z-axis in the air to adjust the landing orientation. To verify the proposed spinal electromechanical system's ability to flexibly mimic natural organisms' spinal structures and behaviors, both single-SGb-SMMS and double-



(a1) (a2) (a3) (a4)

(a) The process of the spine rotating around the x-axis.

(b1) (b2) (b3) (b4)

(b) The process of the spine rotating around the y-axis.

(c1) (c2) (c3) (c4)

(c) The process of the spine rotating around the z-axis.

(d) The motion trajectory diagram of the single-SGb-SMMS robot flipping after falling from the height of 2.8 m.

Fig. 12: Flow schematic illustrations for the flexibility experiment and the aerial flipping experiment

SGb-SMMS robots shown in Fig. 10 are tested for flipping to the target state under various initial conditions.

The experiment has two parts: 1) A single-SGb-SMMS robot is dropped from 2.8 m with initial roll angles of $[90°, 135°, 180°, -135°, -90°]$. Our algorithm enables it to quickly flip to a legs-down posture around the x-axis in the shortest time; 2) A double-SGb-SMMS robot is dropped from 2.8 m and rotates to the target posture around the x-axis and z-axis respectively.

As the robot dropped from 2.8 m lands in 0.8 s, it must flip to the target posture within this time. Figs. 12(d) and 13(a) show that a single-SGb-SMMS robot can fall from 2.8 m and land on all four feet, autonomously planning the shortest path to the legs-down posture from various initial roll angles. Figs. 13(a) and 13(a) indicate that the robot's rotation during flipping adheres to the LOCAM, with the controller maintaining J_i at its maximum of 0.4021. The robot can execute a 180° flip from a back-down posture to a legs-down one within 0.6 s, confirming that the proposed control algorithm emulates falling-cat motion effectively.

Fig. 13(b) shows the attitude-angle changes of the double-SGb-SMMS robot rotating around the body's x-axis and the z-axis in the air. With an initial roll angle of 150°, the double-SGb-SMMS robot can flip around the x-axis to a four-legs-down posture and rotate around the z-axis unaided, shifting the x-axis orientation from 0° to 48°. This verifies that the proposed spinal system can achieve multi-DoF flips via structural expansion, and the control algorithm enables multi-axis flips. Therefore, the cat-falling experiment verifies that the proposed spinal electromechanical system: 1) has the flexibility to actively minic biological dynamic behaviors; 2) has strong expandability in flexible tasks. It is proved that in practical applications, when a quadruped robot or other robots with the SGb-SMMS are dropped in vertical and horizontal initial states, they can adjust multiple postures in the air through our method proposed to achieve a stable landing.

TABLE II: The success rates of different methods in performing the aerial flipping task.

methods	GNN-MPC	Jeffrey's[14]	MPC
success rates	95 %	40 %	< 5 %

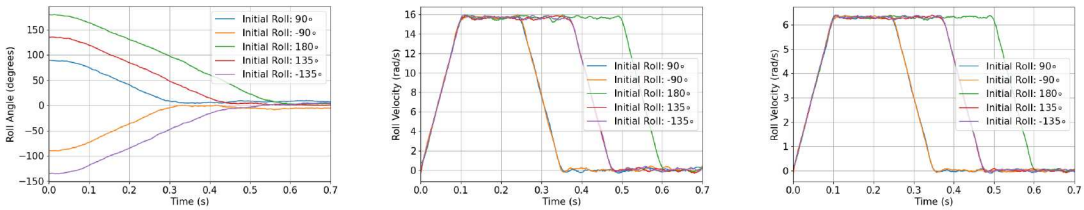
E. Analysis of robustness experiment

In order to verify the robustness and effectiveness of the proposed control algorithm, we used a single-SGb-SMMS robot to conduct multiple comparison and repeated experiments. The robustness experiment has two parts: 1) When given an initial horizontal and yaw angular velocity, the robot can still flip to the target posture before landing; 2) Multiple cat-falling experiments compare our proposed GNN-MPC algorithm with the reduced-order model (Jeffrey's)[14] and MPC without dynamic constraints.

As Fig. 13(c) shows, when dropped from a height of 2.8 m with back-down: 1) With a random initial horizontal velocity, the robot plans an optimal trajectory and flips to a legs-down posture in 0.50 s for a safe landing; 2) With a random initial yaw angular velocity, it follows the desired path and flips to a legs-down posture in 0.70 s. This confirms the robustness of our proposed control algorithm across various conditions.

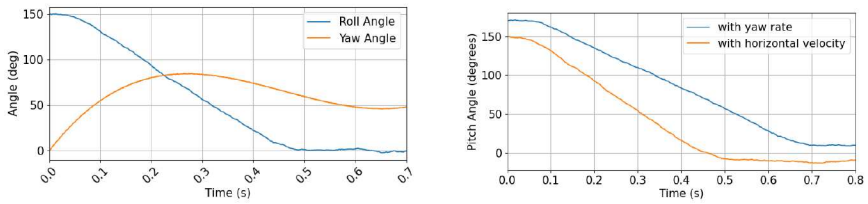
The success rates of the three algorithms in 20 experiments are shown in Table. II. It can be seen that the control algorithm proposed in this paper has a success rate of 95 %, while the algorithm of reduced-order model in [14] only has a 40 % probability of successful flipping. If dynamic constraints are not considered, it is difficult to achieve flipping.

Our proposed control algorithm fully exploits the spine's dynamic performance, allowing the spine-equipped robot to land safely from various initial states, proving its robustness. Repeated tests show it has a higher success rate than algorithms ignoring dynamic constraints, validating its effectiveness.



(a1) The roll angle (a2) The ω_{mass} (a3) The ω_B

(a) The system state of the single-SGb-SMMS robot's flipping under different roll angles.



(b) The angle of the multi-SGb-SMMS robot attitude during flipping.

(c) The roll angle of the robot's flipping under different initial conditions.

Fig. 13: Comprehensive experimental results of the single-SGb-SMMS and multi-SGb-SMMS robots.

VI. Limitations

1. The SGb-SMMS is constructed using 7075-O aluminum alloy with a hardness of approximately 80 HV—significantly lower than that of commonly used gear steels (500–700 HV). After 100 flipping experiments, noticeable wear appears on the gears, resulting in an increased positional error of 0.01 m at the joint ends. Replacing the material with a harder alternative can effectively improve the spine's durability.

2. Currently, the SGb-SMMS is only applied in simplified spinal robots to verify the effectiveness of the control algorithm rapidly. In future work, we will integrate the movement of SGb-SMMS with that of other joints such as legs to further validate the effectiveness of the structure and the algorithm.

VII. Conclusion

This paper proposes a novel spine SGb-SMMS that can achieve omnidirectional continuous three-dimensional bending with only two motors for driving. The proposed GNN-MPC algorithm learns the dynamic parameters offline through a GNN and updates the dynamic constraint online for MPC, thus realizing agile motion control of the spine in complex environments. Through double verification of basic experiments and advanced experiments, the flexibility, strength, expandability, and control precision of the designed SGb-SMMS are verified. The spine can complete multi-posture aerial flipping only by its own joint torques, which verifies the synergistic optimization effect of the structure and the control algorithm. The collaboration of SGb-SMMS and GNN-MPC suits robots of various sizes and has wide applications like speeding up running and navigating complex, confined spaces. Future work will study spinal dynamic constraints to fully utilize its flexibility.

Acknowledgments

This work was supported by the China National Key R&D Program 2022YFB3903804.

References

[1] Christian Aarset, Martin Holler, and Tram Thi Ngoc Nguyen. Learning-informed parameter identification in nonlinear time-dependent pdes. *Appl. Math. Optim.*, 88 (3), August 2023.

[2] Kazuki Abe, Kenjiro Tadakuma, and Riichiro Tadakuma. Abenics: Active ball joint mechanism with three-dof based on spherical gear meshings. *IEEE Transactions on Robotics*, 37(5):1806–1825, 2021.

[3] Michel Aractingi, Pierre-Alexandre Léziart, Thomas Flayols, Julien Perez, Tomi Silander, and Philippe Souères. Controlling the solo12 quadruped robot with deep reinforcement learning. *scientific Reports*, 13(1): 11945, 2023.

[4] Elena Arcari, Maria Vittoria Minniti, Anna Scampicchio, Andrea Carron, Farbod Farshidian, Marco Hutter, and Melanie N Zeilinger. Bayesian multi-task learning mpc for robotic mobile manipulation. *IEEE Robotics and Automation Letters*, 8(6):3222–3229, 2023.

[5] Guillaume Bellegarda, Chuong Nguyen, and Quan Nguyen. Robust quadruped jumping via deep reinforcement learning. *Robotics and Autonomous Systems*, 182: 104799, 2024.

[6] Shounak Bhattacharya, Abhik Singla, Dhaivat Dholakiya, Shalabh Bhatnagar, Bharadwaj Amrutur, Ashitava Ghosal, Shishir Kolathaya, et al. Learning active spine behaviors for dynamic and efficient locomotion in quadruped robots. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1–6. IEEE, 2019.

[7] Zhenshan Bing, Alex Rohregger, Florian Walter, Yuhong Huang, Peer Lucas, Fabrice O Morin, Kai Huang, and Alois Knoll. Lateral flexion of a compliant spine improves motor performance in a bioinspired mouse robot. *Science Robotics*, 8(85):eadg7165, 2023.

[8] Robert Bogue. The role of robots in entertainment. *Industrial Robot: the international journal of robotics research and application*, 49(4):667–671, 2022.

[9] Thomas Buschmann, Sebastian Lohmeier, and Heinz Ulbrich. Humanoid robot lola: Design and walking control. *Journal of physiology-Paris*, 103(3-5):141–148, 2009.

[10] Andrea Carron, Elena Arcari, Martin Wermelinger, Lukas Hewing, Marco Hutter, and Melanie N Zeilinger. Data-driven model predictive control for trajectory tracking with a robotic arm. *IEEE Robotics and Automation Letters*, 4(4):3758–3765, 2019.

[11] Hui Chai, Yibin Li, Rui Song, Guoteng Zhang, Qin Zhang, Song Liu, Jinmian Hou, Yaxian Xin, Ming Yuan, Guoxuan Zhang, et al. A survey of the development of quadruped robots: Joint configuration, dynamic locomotion control method and mobile manipulation approach. *Biomimetic Intelligence and Robotics*, 2(1): 100029, 2022.

[12] Dongliang Chen, Ningjie Li, Hong Wang, and Lei Chen. Effect of flexible spine motion on energy efficiency in quadruped running. *Journal of Bionic Engineering*, 14 (4):716–725, 2017.

[13] Pan Cun-Yun, Wen Xi-Sen, Yang Kun-Yu, Xu Xiao-Jun, Liu Min, and Yao Qi-Shui. Research on transmission principle and kinematics analysis for involute spherical gear. *FRONTIERS OF MECHANICAL ENGINEERING*, 1(2):183–193, JUN 2006.

[14] Jeffrey Duperret and Daniel E Koditschek. Empirical validation of a spined sagittal-plane quadrupedal model. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1058–1064. IEEE, 2017.

[15] Farbod Farshidian, Michael Neunert, Alexander W Winkler, Gonzalo Rey, and Jonas Buchli. An efficient optimal planning and control framework for quadrupedal locomotion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 93–100. IEEE, 2017.

[16] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[17] Rogério Sales Gonçalves and João Carlos Mendes Carvalho. Review and latest trends in mobile robots used on power transmission lines. *International Journal of Advanced Robotic Systems*, 10(12):408, 2013.

[18] S. Gracovetsky. An hypothesis for the role of the spine in human locomotion: A challenge to current thinking. *Journal of Biomedical Engineering*, 7(3):205–216, 1985. ISSN 0141-5425.

[19] Xinyang Gu et al. Advancing Humanoid Locomotion: Mastering Challenging Terrains with Denoising World Model Learning. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.

[20] Liu Hu ran. A new kind of spherical gear and its application in a robot's wrist joint. *Robotics and Computer Integrated Manufacturing*, 25(4-5):732–735, 2009.

[21] Yuhong Huang, Zhenshan Bing, Florian Walter, Alex Rohregger, Zitao Zhang, Kai Huang, Fabrice O Morin, and Alois Knoll. Enhanced quadruped locomotion of a rat robot based on the lateral flexion of a soft actuated spine. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2622–2627. IEEE, 2022.

[22] Cengiz Kahraman, Muhammet Deveci, Eda Boltürk, and Seda Türk. Fuzzy controlled humanoid robots: A literature review. *Robotics and Autonomous Systems*, 134: 103643, 2020.

[23] Yuriko Kakehashi, Kei Okada, and Masayuki Inaba. Development of continuum spine mechanism for humanoid robot: Biomimetic supple and curvilinear spine driven by tendon. In *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 312–317. IEEE, 2020.

[24] Donghyun Kim, Jared Di Carlo, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. *arXiv preprint arXiv:1909.06586*, 2019.

[25] Sunwoo Kim, Maks Sorokin, Jehee Lee, and Sehoon Ha. Human motion control of quadrupedal robots using deep reinforcement learning. In K Hauser, D Shell, and S Huang, editors, *ROBOTICS: SCIENCE AND SYSTEM XVIII*, 2022.

[26] Young Kook Kim, Woojin Seol, and Jihyuk Park. Biomimetic quadruped robot with a spinal joint and optimal spinal motion via reinforcement learning. *Journal of Bionic Engineering*, 18(6):1280–1290, 2021.

[27] David Felipe Leguizamo, Hsin-Jung Yang, Xian Yeow Lee, and Soumik Sarkar. Deep reinforcement learning for robotic control with multi-fidelity models. *IFAC-PapersOnLine*, 55(37):193–198, 2022.

[28] Nian Li, Jialu Cao, and Yiming Huang. Fabrication and testing of the rescue quadruped robot for post-disaster search and rescue operations. In *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*, pages 723–729, 2023.

[29] Wangyang Li, Li Jiang, Ming Cheng, Jinghui Dai, Shaowei Fan, and Hong Liu. Multisensory integrated dexterous finger with coupled-adaptive features. *IEEE/ASME Transactions on Mechatronics*, 2024.

[30] Wanyue Li, Zida Zhou, and Hui Cheng. Dynamic locomotion of a quadruped robot with active spine via model predictive control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1185–1191, 2023.

[31] Xing Li, Xiaofeng Wang, and Jianhui Wang. A kind of lagrange dynamic simplified modeling method for multi-dof robot 1. *Journal of Intelligent & Fuzzy Systems*, 31 (4):2393–2401, 2016.

[32] Guanqi Liang, Lijun Zong, and Tin Lun Lam. Disg: Driving-integrated spherical gear enables singularity-free

full-range joint motion. *IEEE Transactions on Robotics*, 2023.

[33] Zhaopeng Lin, Shiyu Zhou, Zheng Pan, Shaoxun Liu, and Rongrong Wang. Exploring the effect of pitching and rolling spine angle for a galloping quadruped robot via trajectory optimization. In *2023 6th International Conference on Robotics, Control and Automation Engineering (RCAE)*, pages 56–62. IEEE, 2023.

[34] Maria Silva Lopes. Quadruped manipulator for potential agricultural applications. 2023.

[35] Thomas William Mather and Mark Yim. Modular configuration design for a controlled fall. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5905–5910. IEEE, 2009.

[36] Federico Montagnani, Marco Controzzi, and Christian Cipriani. Preliminary design and development of a two degrees of freedom passive compliant prosthetic wrist with switchable stiffness. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 310–315. IEEE, 2013.

[37] Michael O'Connell, Guanya Shi, Xichen Shi, Kamyar Azizzadenesheli, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics*, 7(66): eabm6597, 2022.

[38] Cunyun Pan and Senxi Wen. Structure and kinematic analysis of a flexible wrist for robots based on involute spherical gears. *Journal of Mechanical Engineering*, 41 (7):141–146, 2005.

[39] Andrew P Sabelhaus, Abishek K Akella, Zeerek A Ahmad, and Vytas SunSpiral. Model-predictive control of a flexible spine robot. In *2017 american control conference (acc)*, pages 5051–5057. IEEE, 2017.

[40] Moses Gladson Selvamuthu, Kazuki Abe, Kenjiro Tadakuma, and Riichiro Tadakuma. Metal abenics: Metallic spherical gear mechanism with orientation correction using embedded imu sensor. In *2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 256–263. IEEE, 2024.

[41] Carmelo Sferrazza, Dun-Ming Huang, Fangchen Liu, Jongmin Lee, and Pieter Abbeel. Body transformer: Leveraging robot embodiment for policy learning. *arXiv preprint arXiv:2408.06316*, 2024.

[42] Qing Shi, Chang Li, Kang Li, Qiang Huang, Hiroyuki Ishii, Atsuo Takanishi, and Toshio Fukuda. A modified robotic rat to study rat-like pitch and yaw movements. *IEEE/ASME Transactions on Mechatronics*, 23(5):2448–2458, 2018.

[43] Deyu Su, Shanming Luo, and Jian Wang. Tooth contact analysis of micro-segment spherical gear drives. *Mechanical Science and Technology for Aerospace Engineering*, 34(12):1830 – 1837, 2015. doi: 10.13433/j.cnki.1003-8728.2015.1205.

[44] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*, 2018.

[45] Lingling Tang, Dingkun Liang, Guang Gao, Xin Wang, and Anhuan Xie. Modeling and reinforcement learning-based locomotion control for a humanoid robot with kinematic loop closures. *Multibody System Dynamics*, pages 1–27, 2024.

[46] Li Ting and Pan Cunyun. On grinding manufacture technique and tooth contact and stress analysis of ring-involute spherical gears. *Mechanism and Machine Theory*, 44(10):1807–1825, 2009.

[47] Vassilios Tsounis, Mitja Alge, Joonho Lee, Farbod Farshidian, and Marco Hutter. Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(2): 3699–3706, 2020.

[48] Shengjie Wang, Qing Shi, Junhui Gao, Yuxuan Wang, Fansheng Meng, Chang Li, Qiang Huang, and Toshio Fukuda. Design and control of a miniature quadruped rat-inspired robot. In *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 346–351. IEEE, 2019.

[49] Wei Wang, Aihong Ji, Poramate Manoonpong, Huan Shen, Jie Hu, Zhendong Dai, and Zhiwei Yu. Lateral undulation of the flexible spine of sprawling posture vertebrates. *Journal of Comparative Physiology A*, 204(8):707–719, AUG 2018. ISSN 0340-7594. doi: 10.1007/s00359-018-1275-z.

[50] Xinyue Wang, Jinkun Liu, Ting Zhang, and Jie Zhang. Applied research of agricultural quadruped robots. In *2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)*, pages 954–957. IEEE, 2024.

[51] Hsueh-Cheng Yang. Using an imaginary planar rack cutter to create a spherical gear pair with continue involute teeth. *Arabian Journal for Science and Engineering*, 42: 4725–4735, 2017.

[52] Shyue-Cheng Yang. A rack-cutter surface used to generate a spherical gear with discrete ring-involute teeth. *The International Journal of Advanced Manufacturing Technology*, 27:14–20, 2005.

[53] Qingfeng Yao, Jilong Wang, Shuyu Yang, Cong Wang, Hongyin Zhang, Qifeng Zhang, and Donglin Wang. Imitation and adaptation based on consistency: A quadruped robot imitates animals from videos using deep reinforcement learning. In *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1414–1419. IEEE, 2022.

[54] Qishui Yao, Changyi Li, and Cunyun Pan. Calculation of the conjugate - radius and contact strength at the contact point of the involute spherical gear pair. *Mechanical Science and Technology*, 26(1):96 – 100, 2007.

[55] Keran Ye, Kenneth Chung, and Konstantinos Karydis. A novel lockable spring-loaded prismatic spine to support agile quadrupedal locomotion. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems*

(IROS), pages 2776–2783. IEEE, 2023.

[56] Yevgeniy Yesilevskiy, William Yang, and C David Remy. Spine morphology and energetics: how principles from nature apply to robotics. *Bioinspiration & biomimetics*, 13(3):036002, 2018.

[57] Jie Zhang, Xinyue Wang, and Liang Zheng. Research on autonomous navigation system of agricultural quadruped robot. In *2024 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1286–1290. IEEE, 2024.

[58] Nana Zhu, Hongbin Zang, Bing Liao, Huimin Qi, Zheng Yang, Mingyang Chen, Xin Lang, and Yunjie Wang. A quadruped soft robot for climbing parallel rods. *Robotica*, 39(4):686–698, 2021.

[59] Bi Zhuming, Huang Xinyan, and Li Xiaoning. Quasi-ellipsoidal gear drive used in the flexible joint of robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1300–1305. IEEE, 1996.

APPENDIX

A. Details about Key Physical Parameters

The structural parameters of the single-SGb-SMMS robot experiment platform are shown in Table. III. The maximum bending amplitude θ_s of each joint is $30°$, and the maximum bending amplitude of the entire spine is $180°$.

TABLE III: The structural parameters of the experiment platform

Parameter	Value
Weight	3.272kg
Size	$502 \times 266 \times 160\text{mm}^3$
Spherical Gears-Module	2
Spherical Gears-Tooth	22
Spherical Gears-Material	7075 Al Alloy
Number of Joints	6
Length of Links	44mm

B. Design of Graph Neural Network (GNN)

The input to the GNN consists of point information related to the robot's joints and edge information representing the connection relationships between the joints. The point information includes the bending angle θ_s of the spinal joint and the rotation angle α. We represent each joint as a node in the graph. For a spinal robot with n SGb-SMMS, we have a set of points $P = \{p_1, p_2, \cdots, p_n\}$, where each point p_i is associated with a feature vector $\{\alpha_i, \theta_{si}\}$ of i-th SGb-SMMS.

The edge information is a matrix E that represents the connection relationships between the joints. If joint i and joint j are connected, then $E_{ij} = 1$; otherwise, $E_{ij} = 0$. The GNN consists of multiple layers, each layer performing message passing and feature aggregation.

In the first layer, for each node i, we compute the message from its neighboring nodes j using the following formula:

$$h_{i,j}^1 = \phi^1([h_i^0, h_j^0, E_{ij}, \alpha_j - \alpha_i, \theta_{sj} - \theta_{si}]), \tag{28}$$

where h_i^0 and h_j^0 are the initial features of nodes i and j respectively, and ϕ^1 is a multi-layer perceptron (MLP). The input includes the features of the nodes, the edge connection information, and the differences in joint bending and rotation angles between the neighboring nodes. This helps in capturing the local relationships between the joints.

After computing the messages, we aggregate them for each node i using an aggregation function $\mathcal{A}$ (max function):

$$h_i^1 = \mathcal{A}(h_{i,j}^1 | j \in \mathcal{N}_i), \tag{29}$$

where $\mathcal{N}_i$ is the set of neighboring nodes of i.

We repeat this process for multiple layers. In each subsequent layer k, the message computation formula is:

$$h_{i,j}^k = \phi^k([h_i^{k-1}, h_j^{k-1}, E_{ij}, \alpha_j - \alpha_i, \theta_{sj} - \theta_{si}]), \tag{30}$$

and the aggregation formula is:

$$h_i^k = \mathcal{A}(h_{i,j}^k | j \in \mathcal{N}_i). \tag{31}$$

Finally, after the last layer L, we use an MLP to predict the moment of inertia parameter $\hat{I}$:

$$\hat{I} = MLP(h_i^L). \tag{32}$$

This designed GNN structure aims to capture the complex relationships between the joint information of the spinal robot and predict the dynamic parameters effectively. Given the dataset $D_k = \{P_1, P_2, \cdots P_n\}$ and the corresponding set of the dynamic parameters D_J, the goal of our reward function is to encourage the GNN to accurately predict the dynamic parameters for a given spine state P_i in the dataset D_k.

We define the reward function R as follows:

$$R = \sum_{i=1}^{n} - |\ \Phi(P_i) - J_i|\,, \tag{33}$$

where $\Phi(P_i)$ is the predicted dynamic parameters by the GNN for the input configuration P_i, and J_i is the true dynamic parameters from the dataset D_J.

The negative sign is used to convert the error into a reward that we want to maximize. By minimizing the difference between the predicted and true values of the moment of inertia, the GNN is trained to better approximate the relationship between the input configurations and the dynamic parameters.

In addition, when considering computational efficiency, integrating a lightweight GNN as a trained constraint with IsaacGym has a negligible impact on computational efficiency. We incorporated the GNN as a trained constraint into the open-source legged robot project [19], training 4096 agents on an NVIDIA RTX 3090 Ti Super GPU. The inclusion of the GNN increased the training time per iteration by less than 5% compared to non-GNN training.