ANIMAL LOCOMOTION

Lateral flexion of a compliant spine improves motor performance in a bioinspired mouse robot

Zhenshan Bing[1]†*, Alex Rohregger[1]†, Florian Walter[1,2]†, Yuhong Huang[1], Peer Lucas[1], Fabrice O. Morin[1], Kai Huang[3,4]*, Alois Knoll[1]*

A flexible spine is critical to the motion capability of most animals and plays a pivotal role in their agility. Although state-of-the-art legged robots have already achieved very dynamic and agile movement solely relying on their legs, they still exhibit the type of stiff movement that compromises movement efficiency. The integration of a flexible spine thus appears to be a promising approach to improve their agility, especially for small and underactuated quadruped robots that are underpowered because of size limitations. Here, we show that the lateral flexion of a compliant spine can promote both walking speed and maneuver agility for a neurorobotic mouse (NeRmo). We present NeRmo as a biomimetic robotic mouse that mimics the morphology of biological mice and their muscle-tendon actuation system. First, by leveraging the lateral flexion of the compliant spine, NeRmo can greatly increase its static stability in an initially unstable configuration by adjusting its posture. Second, the lateral flexion of the spine can also effectively extend the stride length of a gait and therefore improve the walking speeds of NeRmo. Finally, NeRmo shows agile maneuvers that require both a small turning radius and fast walking speed with the help of the spine. These results advance our understanding of spine-based quadruped locomotion skills and highlight promising design concepts to develop more agile legged robots.

INTRODUCTION

The spine is an essential structural element of the musculoskeletal system of every quadruped animal, bearing at any time a combination of compressive, tensile, bending, flexional, and torsional loads imparted by the tendino-ligamentous system. It thereby plays a crucial role in generating many agile motion behaviors (*1*, *2*). Although the motion space between two adjacent vertebrae is rather limited, the stacking of many vertebrae together in the spine provides the latter with flexibility and renders it capable of performing a wide range of motions, such as flexion, extension, and rotation (*3*). Among many motor functions provided by the spine, animals mostly use it to adjust the body posture or perform highly dynamic motions. For example, a mouse can squeeze into tiny holes, as small as its skull, by flattening its body via the extension of the spine and collapsing the ribs to the center of the body (*4*). Cats can jump and run agilely by performing extreme flexion and extension of the spine (*5*). More commonly, legged animals rely highly on the spine to perform turning motions (*6*). Therefore, a flexible spine system replicating its biological counterpart's features has great potential to push the physical performance limit of legged robots.

Although a variety of excellent legged robots have been developed in the past decade, such as the Boston Dynamic Spot (*7*), the MIT Cheetah (*8*), and the ETH ANYmal (*9*), they mainly concentrate on leg-based motor skills that lead to stable walking performances on diverse terrains, such as stairs, tunnels, or wild lands (*10*, *11*). Because of their powerful actuation systems, they can already perform limited dynamic behaviors relying only on their legs. However, using a functional spine will further improve motion capabilities, especially for small quadruped robots that are underpowered because of size limitations. There have been few detailed investigations of legged robots equipped with flexible spines, except for a few examples exploring the functionalities of the spine in simulation (*12*–*16*) or implementing rather simplified spine systems with one or two revolute joints (*13*, *17*, *18*). Boston Dynamics' Cheetah robot has one actuated joint that mimics the spine's flexion in the sagittal plane. This design boosts the robot's speed up to 28.3 miles per hour (*19*). The Bobcat robot developed by EPFL has a central revolute joint acting as the spine that connects the front and rear body of the robot (*13*, *20*–*22*). The Waseda rat robot and its successors use a single direct-drive motor that replicates a flexible spine to mimic the turning behavior of rodents (*23*–*26*). However, the basis of the Waseda rat robot is a series of connected wheels, which works more like a train instead of imitating quadruped locomotion. The Kitty robot showcased a variety of spine-driven behaviors because of its flexible spine (*27*). However, the study of synchronized movement between the spine and limbs is not possible because of the lack of leg actuation. The Gecko robot, inspired by reptilian quadrupeds, has a flexible spine with a single direct-drive joint allowing frontal plane flexion (*28*). Despite its structural simplicity, the robot achieves faster locomotion with an extended stride length for symmetric gaits. Additionally, the gecko-inspired robot Slalom uses a flexible spine to execute energy-efficient climbing on inclined surfaces (*29*). The Salamander robot, based on experiments with real salamanders, features a spinal cord model that generates complex patterns of activity for movements like walking and swimming (*30*–*32*). With this model, researchers generated rhythmic movements for the robot to switch between gaits. Beyond systems implementing motor actuation of the spine, other quadruped robots that leverage elastic materials or pneumatic actuation systems for controlling the spine have also been reported (*33*–*37*), but these mechanisms are either too complex or not practically applicable for real-world tasks.

It is challenging to develop a flexible, compact, and functional spine for small and underactuated quadruped robots for several

[1]Chair of Robotics, Artificial Intelligence and Real-Time Systems, School of Computation, Information and Technology, Technical University of Munich, Boltzmannstrasse 3, 85748 Munich, Germany. [2]Machine Intelligence Lab, Department Engineering, University of Technology Nuremberg, Ulmenstrasse 52i, 90443 Nuremberg, Germany. [3]School of Computer Science and Engineering, Sun Yat-sen University, 510330 Guangzhou, China. [4]Pazhou Lab, 510335 Guangzhou, China.
*Corresponding author. Email: zhenshan.bing@tum.de (Z.B.); huangk36@mail.sysu.edu.cn (K.H.); knoll@in.tum.de (A.K.)
†These authors contributed equally to this work.

reasons. First, developing such a flexible spine is structurally complex, requiring innovative design ideas to integrate it and yet still keep the system compact. Second, because of the complex deformation of a flexible spine, it is difficult to model the kinematics or control its flexion effectively. Third, dedicated motion patterns have to be designed to effectively use the spine to coordinate the legs and spine together to achieve agile motion skills. Leveraging our recent research exploring the design concepts (*38*), we present NeRmo, a biomimetic mouse robot that harnesses a flexible spine for agile motion skills. As an underactuated quadruped robot, NeRmo aims to mimic the structural and functional principles of the mouse and therefore the features of its flexible spine, musculoskeletal actuation systems, and fully functional body. We show the enhanced motion skills brought by the lateral flexion of the spine, namely, static balancing, fast walking, and agile turning. We present the kinematic modeling of NeRmo and the control method of the spine-based functionalities. By leveraging the design and control of its lateral spine-based quadruped locomotion, NeRmo successfully demonstrates agile and fast locomotion gaits, a feature often seen in quadruped animals in nature but not fully integrated by modern legged robots.

RESULTS

A biological mouse uses the lateral flexion of the spine to turn around agilely and walk faster (Fig. 1A). Inspired by the biological mouse, this section presents the mouse robot NeRmo and how it takes inspiration from the functional and structural features of the spine (Fig. 1B) and therefore leverages the lateral flexion of its complex spine to perform faster and more agile locomotion gaits (Fig. 1C).

Robot system overview

NeRmo is a miniature quadruped robot that mimics common rodents' musculoskeletal anatomy, visual appearance, and motion patterns (Fig. 1D). The overview of its mechanical design and actuation system is illustrated in Figs. 1D and 2A. NeRmo is unique for several reasons. First, NeRmo has a fully compliant spine that can perform smooth flexion in both the lateral and sagittal planes (Fig. 2B). Second, NeRmo has compliant knee and elbow joints driven by tendons (Fig. 2, C and D). Third, NeRmo's motor-tendon actuation system mimics the musculoskeletal feature of quadruped animals. The primary physical characteristics and specifications of NeRmo are listed in table S8.

Mechanical design

NeRmo is designed to be lightweight, compliant, and modular and to have a flexible skeleton to connect all the body components together (Fig. 2A). Similar to biological rodents, the main load-bearing elements of NeRmo are the following.

NeRmo's spine greatly resembles the structural and functional features of a biological spinal column, which can perform both lateral and sagittal flexion (Fig. 2B). These two types of bending deformations are supported by four joints each, which are stacked in an alternating order to imitate the ball-and-socket joint of a biological spine. This results in an eight-joint design that is highly flexible and provides solid structural support to bear the weight. This feature also makes NeRmo distinctive among other quadruped robots that have spines composed of simple revolute joints. Instead of using muscles, NeRmo's spine is driven by a tendon-pulley system. Two lateral flexion tendons go through the side holes of each vertebra, connecting to the lateral flexion servo at the one end and fixed by an M1 screw at the other end (Fig. 2A). Therefore, similar to the effect of muscle contraction, the rotation of the servo can pull the tendon and then bend the spine laterally to the left or right. In the same way, the sagittal flexion servo pulls a single flexion tendon to stretch the spine vertically. More details about the mechanical properties of the spine can be found in the Supplementary Materials. Note that our

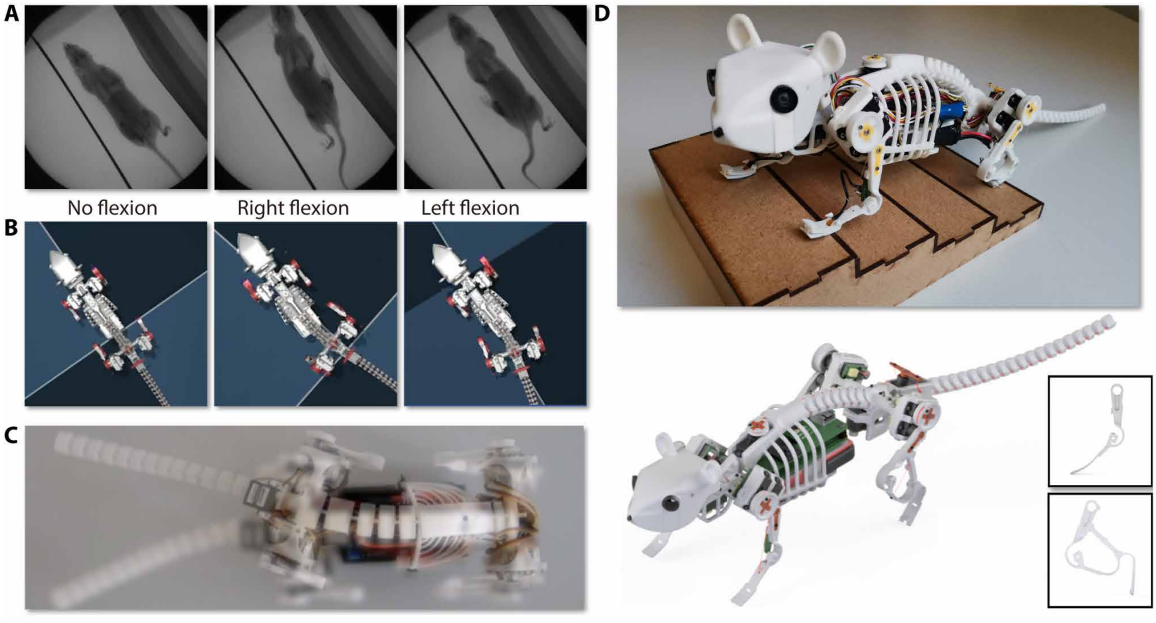

Fig. 1. Snapshots of biological mouse motions and the neurorobotic mouse NeRmo. (**A**) Walking motion of a biological mouse (image courtesy of M. S. Fischer, Institute of Zoology and Evolutionary Research, Friedrich Schiller University Jena). (**B**) NeRmo uses the lateral flexion of the spine for walking in the simulation. (**C**) Lateral flexion pattern of NeRmo's spine. (**D**) NeRmo and its CAD model.

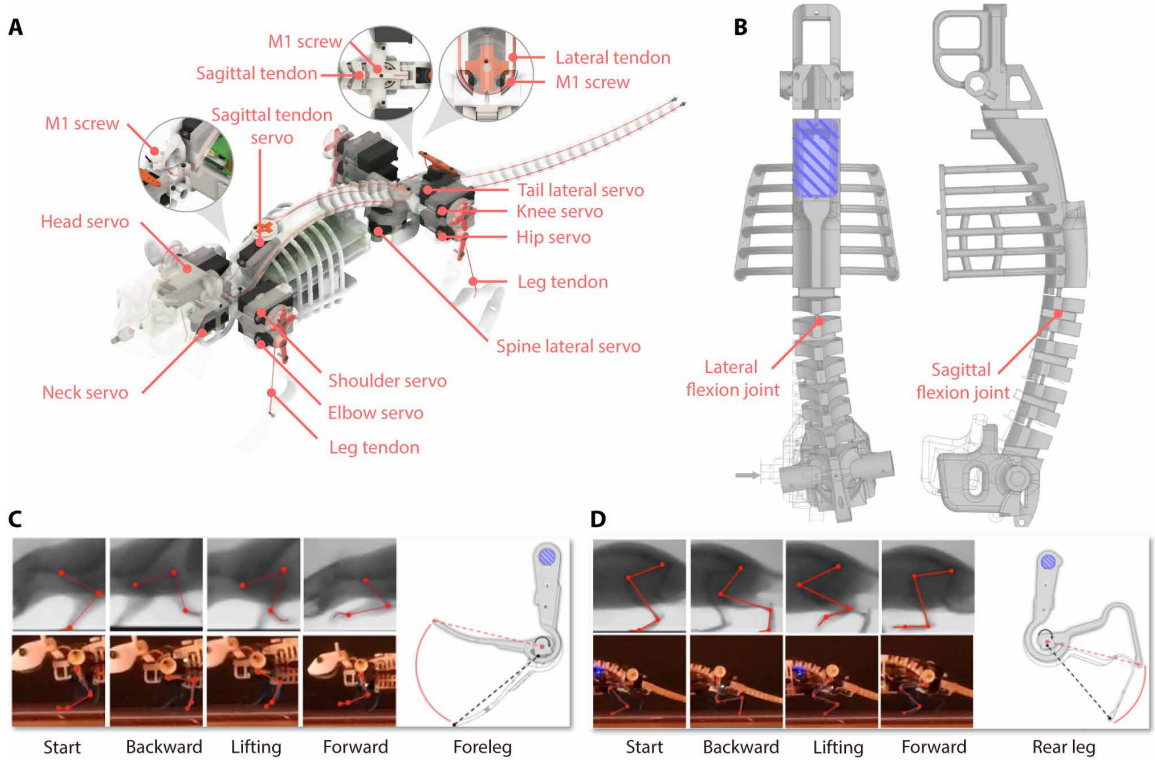

Fig. 2. Structural overview of NeRmo. (**A**) Illustration of servos and tendons used to actuate the flexion of the spine and legs. (**B**) Visualization of the flexion of the spine and the structure of the vertebrae. The wireframe shape depicts the spine's initial configuration before flexion. (**C** and **D**) Visualization of the legs. The wireframe shape shows the initial position of the leg under no tendon compression. Because of the compliance of the material and its mechanical design, the rotation center is aligned with the elbow/knee joint.

work only uses lateral spine flexion to enhance movement; the sagittal flexion is kept fixed, and the tail remains unactuated in all experiments. A discussion about the difference between using a rigid spine versus a fixed compliant spine is given in the Supplementary Materials.

By mimicking the musculoskeletal structure of the rodent's leg, we abstract a three-bar linkage mechanism as NeRmo's foreleg (Fig. 2C) and rear leg (Fig. 2D). The movement of the linkage mechanism closely resembles the motion patterns of rodents' legs with a full cycle of backward, lifting, and forward processes. Both the foreleg and rear leg have a compliant elbow or knee joint and a pseudo-compliant ankle joint. There are several attachment holes to fix the tendon to vary the torque applied to the compliant elbow or knee joint.

Electrical design and software architecture

NeRmo is actuated with 13 Power HD-DSM44 microservos (Fig. 2A). Two servos control the head movement, and one controls a fully compliant tail (similar to the spine). Two servos are used to control the movement of the spine in the dorsal and sagittal planes. Furthermore, each leg also has two servos to control the movement of the shoulder/hip and elbow/knee joints. The onboard computing platform for NeRmo is the Raspberry Pi Zero, which runs the required computations locally, including communication, information processing, and executing motion commands. A modular software architecture built with Robot Operating System (ROS) was implemented to enable a flexible transition between running the controller in simulation and the real world (fig. S14). High-level computations were carried out on a central computer, which also executed the simulation. The control commands were sent to NeRmo via the ROS TCP/IP communication backend. A wireless network enabled the communication between NeRmo and the central computer. Therefore, the modular software architecture lets us control both the simulation and the physical version of NeRmo with the same controller.

Demonstration of locomotion skills

We present four experiments to demonstrate the enhanced locomotion skills, namely, static balancing, straight-line walking, agile turning, and maze navigating. Additional details about the experimental setups are given in the Supplementary Materials.

Static balancing

The static balance of the spine is a physiological alignment of the body posture in the most efficient manner by the muscular forces for mammals. Hence, many disabled quadruped animals can still stand or walk even with three limbs (fig. S15). In the static balancing task, we demonstrated that the spine balance compensation (SBC) function can ensure the static stability of the robot in initially unstable configurations by adjusting the position of its center of mass (CoM) via lateral spine flexion (movies S1 and S2). The test procedures were as follows. First, NeRmo was positioned in a neutral four-foot static contact configuration (Fig. 3A, i) where the leg toes were placed directly under the shoulder or hip. Then, each leg was lifted individually, leading NeRmo into a three-foot static contact

configuration. The lifted leg was controlled to swing in the full motion range of the shoulder or hip joint, with SBC enabled or disabled. NeRmo's standing position and CoM location are depicted in fig. S3. Note that the foreleg lifting was technically an initially stable configuration, because the CoM marginally lay within the supporting triangle formed by the standing legs. Conversely, the rear leg lifting was an initially unstable configuration. The goal was to examine NeRmo's ability to maintain balance and prevent falling over.

In the simulation, NeRmo successfully passed the foreleg lifting tests with or without SBC, because the projection of its CoM still lay in the stability triangle formed by the other three legs. However, lifting the rear left leg without SBC made NeRmo roll over to its left side until it touched the ground (Fig. 3A, ii). In contrast, NeRmo could remained standing when the SBC was switched on (Fig. 3A, iii and iv). NeRmo failed the foreleg balancing tests (with SBC disabled) in the real world, because the legs were more compliant than those in the simulation (Fig. 3B, i). The additional compliance led to more foreleg compression, which shifted the CoM outside the stability triangle, causing NeRmo to fall over. However, NeRmo was still able to keep balance with SBC enabled in this more challenging configuration (Fig. 3B, ii). To quantitatively examine the behavior, we examined the pitch and roll angle of NeRmo in the real world. As shown in Fig. 3C, even when the robot was able to maintain balance, the SBC still reduced the forward pitch by 50% and the roll angle by 50% when lifting up the foreleg. In the rear leg balance test, the SBC reduced the pitch deviation by 73% and roll deviation by 80% (Fig. 3D) and successfully prevented the robot from falling over.

Straight-line walking

In the straight-line walking task, we demonstrated that the walking speed could be increased by using the lateral flexion of the spine (movies S3 to S8). The experimental setups are shown in Fig. 4A. The distance between the starting point and the end was 2.0 m in the simulation and 1.5 m in the real world. To show the general effectiveness of the spine in improving the velocity, we performed the experiments by varying a group of parameters, namely, gait type, frequency, spine actuation, and normalized stride length (table S9). Three types of gaits were tested: the ideal trot, the walking trot, and the lateral sequence walk (*39*). The gait patterns are visualized in fig. S16. According to (*39*), the ideal trot gait had a standing and swing phase of 50% each in one cycle. In the walking trot, the standing phase was approximately 60%, whereas the swing phase was 40%. Another common gait for mice is lateral sequence walking, where all four limbs move in a lateral sequence. The normalized stride length is defined as $l_n = l_{sl}/l_{sl,\,max}$, where l_{sl} is the stride length of the current gait pattern and $l_{sl,\,max}$ is the maximum stride length.

The experiment results depicted in Fig. 4 (B to E) show that the spine-based gaits outperformed normal gaits in terms of velocity. Figure 4C shows the walking speeds under different gait frequencies obtained from the real world (movie S3). Six configurations of gaits were tested, including the ideal trot with or without the spine ("trt,s" or "trt,ns"), walking trot with or without the spine ("walk,s" or "walk,ns"), and the lateral walking gait with or without the spine ("lat,s" or "lat,ns"). Depending on the gait frequency, the speed increment ranged between 8% (for a 0.5-Hz lateral sequence walk) and 17% (for a 0.8-Hz ideal trot) in the real world. An intuitive visualization of the stride length increase is shown in Fig. 4B. We observed that the stride length of the spine-based gait was greater than the standard gait. We also observed that the swing leg trajectory of the spine-based gait had a characteristic curve shape compared with the linear trajectory of the standard gait, which was caused by the spine flexion. For the ideal trot gait at 0.8 Hz, the real-world results in Fig. 4D show that the gait speed increases with the stride length, and the spine flexion was effective in increasing the speed constantly. The real-world results validated the outcomes and trends

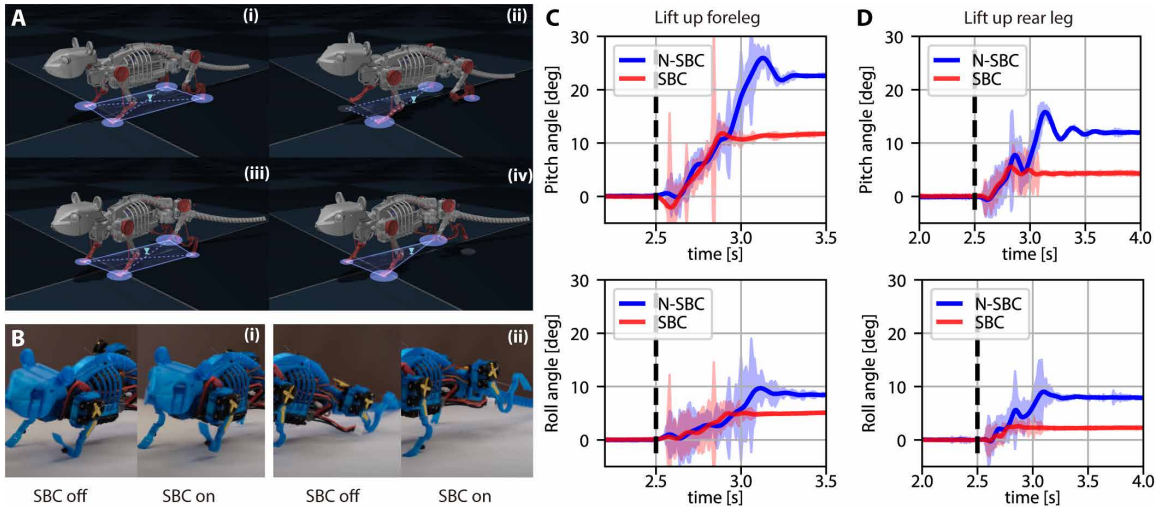

Fig. 3. Experimental results of the static balancing task. (**A**) Static balance test of lifting the left rear leg in the simulation. The blue ellipses qualitatively depict the contact feet and forces (larger ellipse for greater contact force). The triangle depicts the projection of the CoM onto the ground. (i) Four-foot contact configuration without SBC. (ii) Static three-foot contact configuration without SBC (test failed, and NeRmo tilted to its left side until the left rear foot touched the ground). (iii) Static four-foot contact with SBC. (iv) Static three-foot contact with SBC (test passed). (**B**) (i) Lifting the left foreleg in the real world. (ii) Lifting the left rear leg in the real world. In both cases, NeRmo fell over without SBC and kept its balance with SBC enabled. (**C** and **D**) Pitch and roll orientation when lifting legs in the real world. The solid line represents the average of five runs, and the shaded area indicates the variance. The vertical dashed line indicates the time of leg lifting. The balance failure happened when the pitch angle reached the maximum value.

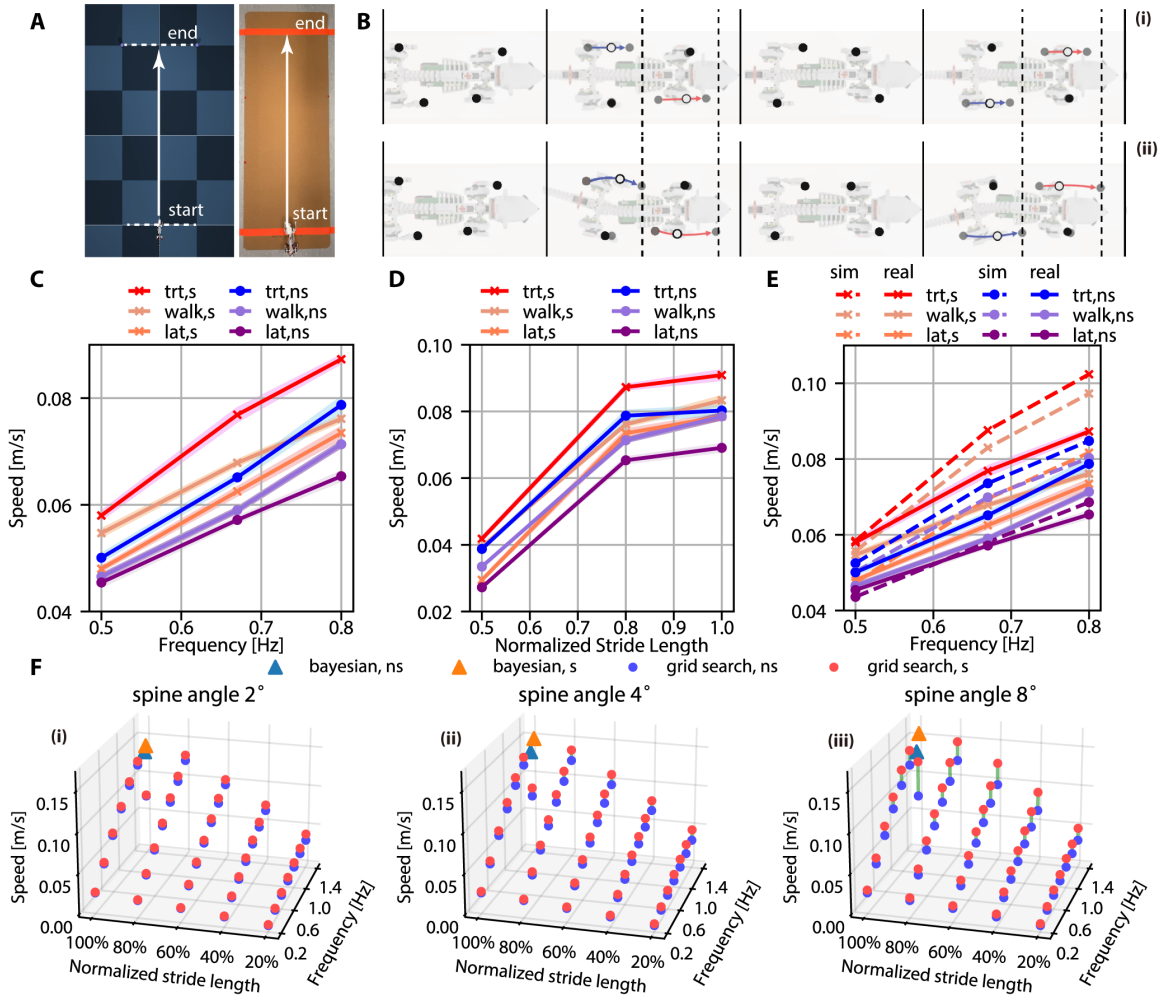

Fig. 4. Experimental results of the straight-line walking task. (**A**) Testing setup in the simulator and the real world. (**B**) Visualization of the foot placements for an ideal trot across a gait cycle (upper row) and a spine-extended ideal trot (lower row). Solid circles indicate foot placements in the stance phase, hollow circles indicate foot placements in the swing phase, and arrows indicate leg trajectory in the swing phase. (**C**) Real-world results show that the speed is dependent on gait frequency, gait type, and spine actuation. The gait types include lateral sequence walk (lat), walking trot (walk), and ideal trot (trt). The label "ns" denotes the setting without the spine-based stride extension, and "s" denotes the setting with spine-based stride extension. (**D**) Real-world results show that the speed is dependent on the normalized stride length for an ideal trot at 0.8 Hz. (**E**) Comparison of straight-line speeds between gaits in the simulation (dashed lines) and real-world experiments (solid lines are averaged values of five trials, and shadow areas show the variance). (**F**) Speeds of ideal trot gaits generated from the grid search algorithm (blue points, no spine; red points, with spine) and the Bayesian optimization algorithm (blue triangle, no spine; yellow triangle, with spine). Green lines show the improvement between comparison pairs.

of the simulation tests (Fig. 4E and movies S4 to S8). However, there was a speed discrepancy between real-world and simulation results. For example, the real-world speed was around 14% slower for the ideal trot with spine extension at 0.8 Hz. A full comparison between simulation and real-world results is given in fig. S17.

Furthermore, we demonstrated the applicability of the spine's speed improvement across different gait control methods in the simulation. We evaluated the effectiveness of two baseline controllers, the grid search controller and the Bayesian optimization controller, in searching for faster gaits. The grid search controller explored gait parameters by searching a parameter grid at fixed intervals to find the fastest gaits. We then used the Bayesian optimization algorithm to search for optimized parameters within those grid intervals. This approach has been used to optimize gait parameters in various robots, particularly legged robots (*40*–*44*). Implementation details of these two controllers are given in the Supplementary Materials. Figure 4F displays the simulation results of the ideal trot gait. Each scatter represents one gait under a specific combination of parameters. We can observe that the gait speed increased with the walking frequency and the normalized stride length. The Bayesian optimization controller was able to find speed-optimized gaits (triangle marker) with or without the spine compared with the grid search controller (solid points). More importantly, the spine improved the speed of the gait for all parameter combinations in the grid search algorithm. Figure S5 presents the results of the walking trot and lateral sequence walk gait.

Agile turning

In the agile turning test, we demonstrated the enhanced turning capability introduced by the spine (movies S9 and S10). Figure 5A shows the experimental setups, where the robot was directed to turn

along two paths with specified radii. Table S10 lists the independent variables, including gait type, frequency, turning radius, and turning strategy. This study tested three turning strategies: the leg-based turning, spine-based turning, and mix-based turning. The leg-based turning strategy uses varying stride lengths on each side to steer the robot's movement, creating an asymmetric contact force distribution that produces a yaw moment around the CoM (fig. S9). Note that the leg-based turning strategy discussed in this study is applicable only to legs without abduction or adduction actuation. For legs with such actuation, turning can be achieved by laterally positioning the legs, introducing a yaw movement directly (*45*). The spine-based turning strategy uses lateral spine flexion to steer the robot, similar to biological behavior (Fig. 5B). The mix-based turning strategy combines both leg-based and spine-based strategies to maneuver the robot.

The real-world results presented in Fig. 5C demonstrated that spine-based lateral flexion turning (denoted by "sb") was the fastest across all gait frequencies and types regardless of the turning trajectory, which achieved around 20% to 30% faster traversal speed than leg-based turning. The mix-based turning (denoted by "mb") strategy also achieved faster speed than leg-based turning (denoted by "lb") and was only slightly slower than the spine-based turning strategy. For example, for the ideal trot and walking trot gait with a turning radius of 1.0 m at 0.8 Hz, the difference between the spine and mix-based turning was negligible, at around 3%. Although the spine-based turning increases the turning speed compared with

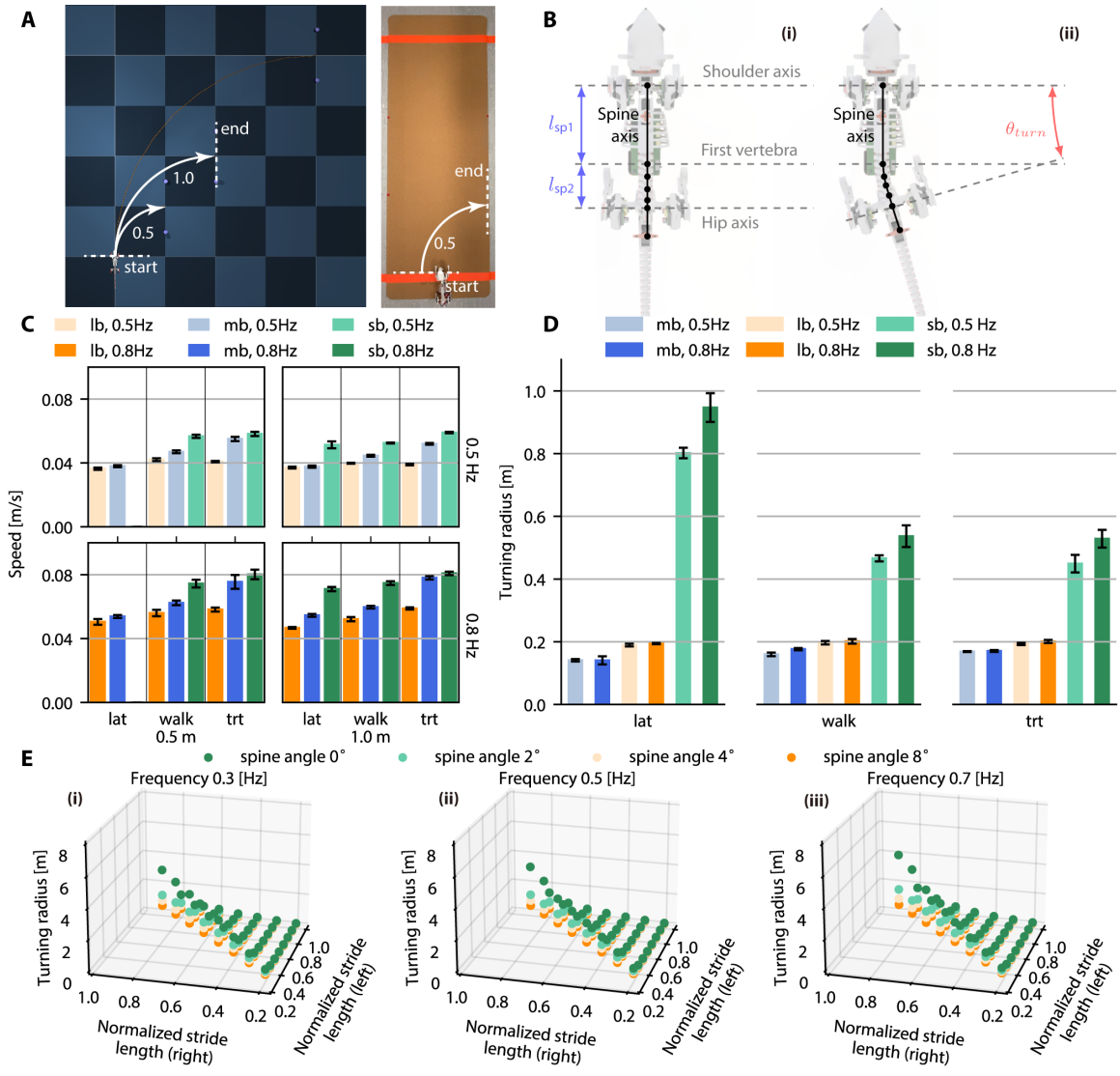

Fig. 5. Experimental results of the turning task. (**A**) Testing setup in the simulator and the real world. (**B**) Illustration of the spine-based turning strategy. (i) The shoulder and hip axes are parallel, with no turning radius induced by the spine. (ii) Spine-based lateral flexion turning with turning angle θ_{turn}. (**C**) The turning strategies are mix-based turning (mb), leg-based turning (lb), and spine-based (sb) turning. Real-world turning speeds with different gaits and turning strategies across the 0.5-m- and 1.0-m-radius trajectory. Error bars represent SD from five trials. (**D**) Real-world minimum turning radii achieved by different gait types, frequencies, and turning strategies. Error bars represent SD from five trials. (**E**) Simulation results of the minimum turning radius of ideal trot gaits generated from the grid search algorithm.

other turning strategies, its minimum achievable turning radius was limited by the flexion range of the spine. The lateral sequence walking ("lat") failed to turn with a radius of 0.5 m at both 0.5 and 0.8 Hz. Therefore, no histogram is provided in Fig. 5C to depict these results. The results also showed that the mix-based turning strategy could compensate for the limited turning radius of the spine-based strategy and still achieve a fast-walking speed.

As shown in Fig. 5D, the mix-based turning strategy achieved the best minimum turning radius in the real world (0.14 m), which was around 30% lower than the minimum turning radius achieved by leg-based turning and 60% to 85% smaller than the spine-based turning gaits. The simulation results are shown in fig. S18. There was a roughly 10% to 20% difference in the results between the simulation and real-world tests. It can be concluded that the trends from the simulation for the turning behavior were transferable to the real world but with a loss of expected performance.

Similar to the walking experiment, we also implemented the grid search controller to search for turning-optimized gaits (see the Supplementary Materials). Figure 5E illustrates the simulation results of ideal trot gaits achieving the minimum turning radius for various parameter combinations. Each scatter point represents a gait obtained from the parameter space search. Gaits with equal stride lengths for the left and right legs correspond to the spine-based turning strategy, where the legs were not actively involved in the turning process. Gaits labeled as "spine angle 0°" represent the leg-based turning strategy, and the remaining gaits indicate a mixed-based turning approach. Observing the results, we can see that the lateral flexion of the spine consistently achieved a smaller turning radius across all settings. Furthermore, a larger flexion angle of the spine correlated with a smaller turning radius, and the minimum turning radius tended to increase with the walking frequency of a gait. Figures S6 and S7 present the full results of three types of gaits.

Maze navigating

To evaluate the gait performance that combines both straight-line walking and turning motion, we conducted the maze navigating task in a maze environment. The dimensions of the maze and the simulation environment are visualized in Fig. 6 (A and B), and the real-world maze setup shared the same dimension and is shown in fig. S10. The robot was tested in two configurations, namely, with the spine controller enabled and disabled. For the first configuration, the robot was controlled to perform mix-based turning for the curved path and spine-based walking for the straight-line path. For the latter configuration, the robot was controlled to perform only leg-based motion. As shown in Fig. 6A, the maze scenario required the robot to perform a series of walking and turning to navigate through the maze. To determine the path that NeRmo should follow, a vision sensor was used to acquire visual data. Thereafter, a high-level controller decided whether the robot should move forward, turn right, or turn left to navigate and follow the desired path. By implementing this approach, we could ensure that NeRmo traveled the same distance in the maze, allowing for a fair comparison of the travel time for different gait types.

The most profound influence of the spine functionalities on locomotion performance was observed in the maze scenario simulations (see Table 1 and movie S11). With spine-based stride extension and mix-based turning, the simulation completed the maze in 50.5% less time and achieved a 105% faster average speed through the maze. In the real-world tests (Fig. 6, C and D, and movie S12), the robot took

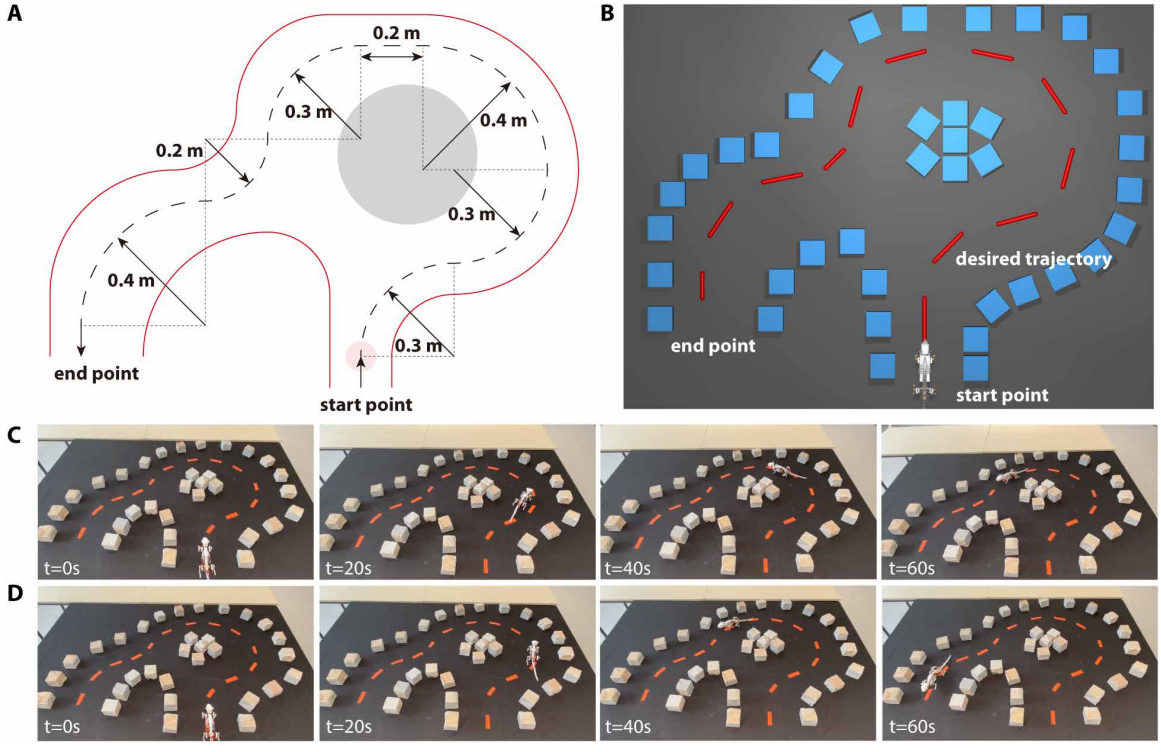

Fig. 6. Experimental results of the maze navigating task. (**A**) Sketch of the maze and (**B**) the simulation maze. The overall length of the desired maze was about 3.46 m. (**C**) NeRmo navigated in the maze without spine-based functions. (**D**) NeRmo navigated in the maze with spine-based functions (turning and straight-line walking).

30.6% less time to complete the maze when the spine was activated. In summary, the simulation and real-world results confirmed the performance increase of the spine-based functionalities demonstrated in the straight-line and turning tests.

DISCUSSION

Motivated by studying the cognition and motion principles of mammals, we developed the bioinspired robotic mouse NeRmo, which mimics the major musculoskeletal features of mice that enable efficient quadruped locomotion. The most important feature of NeRmo is its compliant and flexible spine, with which it can perform several spine-based motion skills.

NeRmo's spine mimics the physical structure of a biological spine, using nylon's tensile strength and flexibility to bear its body weight and maintaining extension and flexion. The spine has two segments, with the front being rigid and responsible for housing all the electronic devices. The back mimics vertebrae and becomes flexible by stacking four lumbar and lateral flexion joints alternately, with two tendons bending it for lateral flexion. Unlike NeRmo, most quadruped robots acquire the flexion of a rigid spine via only one or two simplified revolute joints that connect the front and rear body parts. There are also some quadruped robots that achieve the flexibility of the nonrigid spine via complex mechanisms or rubber materials, such as tensegrity (*18*), the spring-linkage mechanism (*46*), and the elastic rod mechanism (*47*). NeRmo's spine presents a flexible, compact, and simple design concept for robotic spines, with the potential for future legged robots to use additive manufacturing for developing complex yet flexible body parts.

We successfully demonstrated that a flexible spine could improve the locomotion performance of small underactuated quadrupeds in three areas: static stability, walking speed, and turning behavior. We demonstrated that the spine improved the static stability of quadrupeds without foot repositioning. With the SBC enabled, NeRmo remained balanced in three-foot contact scenarios that were initially statically unstable. This approach can increase the robustness of quadruped robots in real-world applications and potentially enable them to remain functional in the event of leg malfunction. We show that the lateral flexion of a spine can enable faster walking speeds by extending the effective stride length compared with leg-based walking. Therefore, spine-based stride extension provides an additional method, beyond increasing the gait frequency, to further increase the gait speed for quadruped robots. Another substantial advantage of the flexible spine is that it enhances the maneuverability of underactuated quadrupeds. State-of-the-art quadruped robots are limited to leg-based turning without an active spine, which is rather inefficient. With the addition of an active spine, we have shown that the turning speed of NeRmo can be increased by up to 40% compared with leg-based turning, depending on the turning radius and gait type. This is because spine-based turning does not need to modulate the leg stride length but takes advantage of the entire forward leg motion while turning. Furthermore, the minimum turning radius can also be improved by up to 30% with the mix-based turning strategy compared with leg-based turning.

Future work could explore quadruped locomotion with a more complex spine and apply it to highly dynamic motions. Here, we only investigated the effect of lateral spine flexion on improving the agility of NeRmo. From research on rodent gaits (*48*), we know that rodents also use vertical flexion of the spine for their gaits. Therefore, extending the spine model by enabling vertical flexion also has great potential to further boost the physical limit of NeRmo or other quadruped robots.

MATERIALS AND METHODS

Kinematic model of spine

NeRmo leverages its kinematics to calculate and coordinate its spine-based motions. Therefore, we present the derivation of the kinematic model of the spine, which is used to generate walking trajectories and to enable spine-based functionalities. The kinematic models of the foreleg and rear leg are given in the Supplementary Materials.

The compliant spine can be modeled as a series of connected, discretized, compliant joints. Figure 7A visualizes the computer-aided design (CAD) model of the spine superimposed with its kinematic wireframe. We assumed that NeRmo's front torso is fixed at the base frame M. The spine was then modeled as an eight-axis manipulator, with the rear torso of NeRmo, namely, the hips, rear legs, and tail, as the tool-end of the manipulator, denoted by the frame {10}. The spine is composed of both sagittal and lateral flexion joints. The four lateral flexion joints are located at frames {2}, {4}, {6}, and {8}, denoted by respective joint angles θ_2, θ_4, θ_6, and θ_8. The sagittal flexion joints are at frames {1}, {3}, {5}, and {7} with their respective joint angles θ_1, θ_3, θ_5, and θ_7. The sagittal joint values are set constant because we only focus on lateral flexion in this work.

We used the Denavit-Hartenberg (DH) convention to compute the forward kinematics of the spine. The modified DH parameters of the spine are listed in table S11. The transformation matrix between the previous vertebra's coordinate frame $i-1$ and the current vertebra's frame i was deduced as

$$ {}^{i-1}\boldsymbol{T}_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & \alpha_{i-1} \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_i) & -d_i\sin(\alpha_{i-1}) \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & d_i\cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

Table 1. Experiment results of the maze navigating test.

	Simulation			Real world (five trials)		
Metrics	**No spine**	**With spine**	**Change**	**No spine**	**With spine**	**Change**
Times taken (s)	81.8	40.5	−50.5%	90.2 (±1.65)	62.6 (±1.77)	−30.6%
Average speeds (m/s)	0.042	0.086	105%	0.038	0.055	44.7%

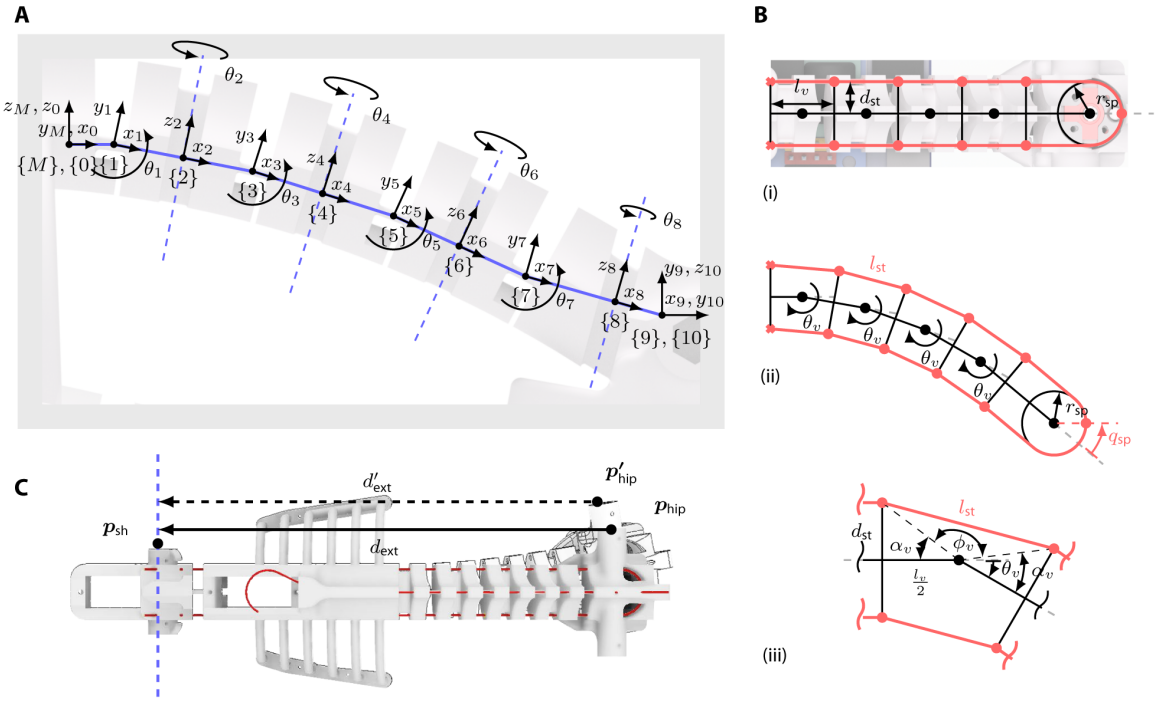

Fig. 7. Kinematics of the spine. (**A**) Kinematic diagram of the spine according to the DH convention with eight revolute joints denoted by θ_i. Frame M denotes the base frame attached to the main torso of NeRmo, and frame 10 denotes the end frame attached to the hip of NeRmo. (**B**) (i) Schematic of the spine in neutral configuration ($q_{sp} = 0$) with the CAD of the spine underneath. (ii) Schematic of the spine with the servo actuated ($q_{sp} \neq 0$). (iii) Kinematic diagram of a single spine element. (**C**) Schematic of the distance d_{ext} between the shoulder $\boldsymbol{p}_{sh}$ and the hip $\boldsymbol{p}_{hip}$ during the lateral flexion.

The complete transformation matrix from the end frame {10} located at the hip to the base frame M of the spine can be expressed as

$$ {}^{M}\boldsymbol{T}_{10} = {}^{M}\boldsymbol{T}_{0} \prod_{i=1}^{10} {}^{i-1}\boldsymbol{T}_{i} \tag{2} $$

where ${}^{M}\boldsymbol{T}_{10}$ denotes the transformation matrix from the end frame {10} to the base frame M. We can represent ${}^{M}\boldsymbol{T}_{10}$ as a function of the lateral joints $\boldsymbol{f}(\theta_2, \theta_4, \theta_6, \theta_8)$. Then, any point ${}^{10}\boldsymbol{p}_e$ in the coordinate frame of the endpoint (hip) can be described as a point in the base frame ${}^{M}\boldsymbol{p}_e$ with

$$ {}^{M}\boldsymbol{p}_e = {}^{M}\boldsymbol{T}_{10} \cdot {}^{10}\boldsymbol{p}_e \tag{3} $$

where ${}^{M}\boldsymbol{p}_e = \boldsymbol{f}(\theta_2, \theta_4, \theta_6, \theta_8)$.

To simplify the kinematic model, we assumed that the angular displacement of each lateral spine joint is identical such that the angular displacement of each joint is expressed by a single variable $\theta_i = \theta_v$ (Fig. 7B, i). This assumption holds because each lateral spine joint is mechanically equal and connected in series. Therefore, the introduced assumption reduces the forward kinematics expressed by the transformation matrix to a single unknown variable ${}^{M}\boldsymbol{T}_{10} = \boldsymbol{f}(\theta_v)$.

Thus far, we have expressed the kinematics of the spine with respect to the spine vertebral angle θ_v. However, the spine is actuated by the tendon system with the spine pulley servo q_{sp} (Fig. 7B, ii). The rotation of the spine servo exerts a torque that is equally distributed to each of the four joints based on the equal joint assumption. The actuation of the spine servo results in a relative change in the spine tendon length l_{st} on either side of the vertebra dependent on q_{sp}. Therefore, the relationship between θ_v and q_{sp} is derived from the diagram of a single spine segment (Fig. 7B, iii) as

$$ \theta_v = 2\alpha_v - \pi + \arccos\left(1 - \frac{l_{st}^2}{2((l_v/2)^2 + d_{st}^2)} \right) \tag{4} $$

$$ l_{st} = l_v + \Delta l_{st} \tag{5} $$

$$ \Delta l_{st} = (r_{sp}(q_{sp} - q_{sp0}))/4 \tag{6} $$

$$ \alpha_v = \arctan \frac{d_{st}}{l_v/2} \tag{7} $$

where d_{st} is the distance of the tendon to the central axis, l_{st} is the effective tendon length between each spine segment, l_v is the distance between two spine vertebra elements, Δl_{st} is the change in effective tendon length, q_{sp} is the actuation angle of the spine servo, q_{sp0} is the neutral position of the spine servo for which $\Delta l_{st} = 0$ (for $q_{sp0} = 0$), and r_{sp} is the radius of the spine servo pulley. The factor of 4 in Eq. 6 is a result of the property that the change in tendon length is distributed equally across all four lateral joints of the spine. Solving Eq. 4 for q_{sp} yields the explicit solution

$$q_{sp} = \begin{pmatrix} \frac{\sigma_2 - 4l_v + \sigma_1 + r_s q_{sp0}}{r_s} \\ -\frac{4l_v + \sigma_2 + \sigma_1 - r_s q_{sp0}}{r_s} \end{pmatrix} \quad (8)$$

where

$$\sigma_1 = 2l_v^2\sigma_3$$
$$\sigma_2 = 8d_{st}^2\sigma_3$$
$$\sigma_3 = \sqrt{-\frac{-2\cos\theta_v \cdot \cos\alpha_v^2 + 2\sin\alpha_v \cdot \sin\theta_v \cdot \cos\alpha_v + \cos\theta_v - 1}{4d_{st}^2 + l_v^2}} \quad (9)$$

Last, we require the inverse kinematic model to compute a solution for θ_v given a desired hip deflection, which is then used to solve for q_{sp} with Eq. 8. Similar to the inverse kinematic models of the legs, the inverse kinematic model of the spine is computed in the velocity space using Eq. 19, where the Jacobian $\boldsymbol{J} = {}^{M}\dot{\boldsymbol{T}}_{10}$. The solution for θ_v is then computed numerically and used to solve for the spine pulley servo q_{sp} with Eq. 8.

Spine-based functionalities

We derive control models for three spine-based functionalities that enhance the locomotion behavior of NeRmo: spine-based balance compensation to improve static stability, spine-based stride extension to increase locomotion speed, and spine-based turning to enhance maneuverability.

Spine-based balance compensation

The lateral spine flexion constantly affects the position of NeRmo's CoM. When the spine is actuated, the front torso is translated relative to the rear torso, resulting in a shift of the CoM. As illustrated in fig. S3, its position $\boldsymbol{p}_{com}$ is derived by considering the rear and front torso as two bodies connected by a massless spine such that

$$\boldsymbol{p}_{com} = \frac{1}{m_{fr} + m_{rr}}\left[m_{fr}\boldsymbol{p}_{com_{fr}} + m_{rr}\left({}^{M}\boldsymbol{T}_{10} \cdot {}^{10}\boldsymbol{p}_{com_{rr}}\right)\right] \quad (10)$$

where m_{fr} and m_{rr} are the masses of the front and rear torso, $\boldsymbol{p}_{com_{fr}}$ denotes the position of the CoM of the front torso, and ${}^{10}\boldsymbol{p}_{com_{rr}}$ denotes the position of the CoM of the rear torso in the reference frame of the hip. The spine is considered massless in the CoM model because its mass constitutes less than 2% of the total mass.

We show the change of the CoM position with respect to the spine vertebral angle θ_v and the spine pulley actuation q_{sp} in Eq. 10. Therefore, $\boldsymbol{p}_{com}$ can be used to derive a model that controls the stability of NeRmo in the three-foot contact configuration. In the three-foot contact scenario (fig. S3), NeRmo can transition between static stability and instability depending on the CoM position. When both rear feet and one front foot are in contact (stable configuration), the CoM lies within the stability triangle enclosed by the stability diagonal. Conversely, NeRmo is statically unstable for cases with two front feet and one rear foot in contact because the CoM is outside the stability triangle. Therefore, actuating the spine to translate the CoM relative to the stability diagonal influences NeRmo's static stability.

The CoM position relative to the stability diagonal, using a distance metric d_{stab}, is defined as

$$|\, d_{stab} \,| = \frac{|\,(\boldsymbol{p}_{fe,y} - \boldsymbol{p}_{re,y})\boldsymbol{p}_{com,x} + (\boldsymbol{p}_{re,x} - \boldsymbol{p}_{fe,x})\boldsymbol{p}_{com,y} + (\boldsymbol{p}_{fe,x}\boldsymbol{p}_{re,y} - \boldsymbol{p}_{fe,y}\boldsymbol{p}_{re,x})\,|}{\sqrt{(\boldsymbol{p}_{fe,y} - \boldsymbol{p}_{re,y})^2 + (\boldsymbol{p}_{re,x} - \boldsymbol{p}_{fe,x})^2}} \quad (11)$$

where $\boldsymbol{p}_{com,x}$ and $\boldsymbol{p}_{com,y}$ denote the x- and y-projection of the CoM onto the ground plane. $\boldsymbol{p}_{fe}$ and $\boldsymbol{p}_{re}$ denote the known front and rear foot-end positions of the opposing feet contact pair defining the stability line. Therefore, the distance d_{stab} can be solved with θ_v and q_{sp} from Eq. 8. The stability distance d_{stab} is a signed function that provides information on the relative orientation between the CoM position and the stability diagonal. For example, if the CoM is located to the right of the stability diagonal, then d_{stab} is greater than zero, whereas on the left of the stability diagonal, d_{stab} is less than zero. The numerical simulation of the CoM position can be seen in movie S1.

Spine-based stride extension

The lateral flexion of the spine translates the rear torso of NeRmo relative to the front torso, which leads to the distance change between the left shoulder and right hip or vice versa (Fig. 7C). We leverage this variation of the relative shoulder-to-hip distance to introduce the spine-based stride extension. The distance d_{ext} between a hip-shoulder pair is computed as

$$d_{ext}(\boldsymbol{p}_{sh_j}, \boldsymbol{p}_{hip_i}) = ||\boldsymbol{p}_{sh_j} - \left({}^{M}\boldsymbol{T}_{10} \cdot {}^{10}\boldsymbol{p}_{hip_i}\right)|| \quad (12)$$

where $i, j \in$ left, right, and $i \neq j$. The point $\boldsymbol{p}_{sh_j}$ denotes the coordinates of the shoulder in the base frame M. $\boldsymbol{p}_{hip_i}$ denotes the coordinates of the hip. Because $\boldsymbol{p}_{sh_j}$ is constant in the base frame, the distance $d_{ext}(\boldsymbol{p}_{sh_j}, \boldsymbol{p}_{hip_i})$ only depends on the bending angle of the spine θ_v. With the distance d_{ext}, we are able to switch between two different modes, spine-based gait extension (increasing the distance) and spine-based gait compression (decreasing the distance). To enable gait extension, we compute a solution for the spine vertebral angle $\theta_v = \theta^*_{v,ext}$ by maximizing Eq. 12 such that

$$\theta^*_{v,ext} = \max_{\theta_v} d_{ext}(\boldsymbol{p}_{sh_j}, \boldsymbol{p}_{hip_j}), \text{ with respect to } 0 \leq \theta^*_{v,ext} \leq 0.1 \text{ rad} \quad (13)$$

$\theta^*_{v,ext}$ is constrained to 0.1 rad because of the mechanical limitation and the safety factor of the spine flexion. Gait compression was achieved by minimizing the distance defined in Eq. 12 to obtain a spine vertebral angle $\theta_v = \theta^*_{v,comp}$.

By synchronizing the spine extension and compression with four legs during a gait cycle, we can increase the effective stride length of a gait. The strategy is applied to symmetric gaits, e.g., the trot, according to two rules. For an opposing leg pair in the swing state, we apply the spine-based gait compression that effectively increases the length of their swing by forward yaw translation. For an opposing leg pair in the stance state, we apply the spine-based gait extension that effectively increases the length of their stance by rearward yaw translation. Figure 4B demonstrates the spine-based stride extension effect by comparing a standard ideal trot gait (Fig. 4B, i) with a spine-extended ideal trot (Fig. 4B, ii).

Spine-based turning strategy

We introduced three control strategies for NeRmo to steer in a direction: the spine-based lateral flexion turning (sb), leg-based turning (lb), and mixed-based turning (mb). The implementation details

of the leg-based turning strategy are given in the Supplementary Materials.

The spine-based turning strategy takes advantage of the lateral spine flexion. The induced lateral flexion displaces the rear legs and changes the yaw orientation by angle θ_{turn}, as illustrated in Fig. 5B. The change of the distance between the shoulder and hip d_{ext} enables a turning motion and yaw moment around the CoM of the robot. Trivially, $\theta_{turn} = 4 \times \theta_v$, where θ_v is the joint angle of the spine vertebrae used for the lateral flexion (see Fig. 7B). l_{sp1} and l_{sp2} are body lengths. Therefore, the turning radius can be derived as

$$r_{turn}(\theta_{turn}) = \frac{l_{sp2}}{\theta_{turn}} + \frac{l_{sp1}}{\tan\theta_{turn}} \quad (14)$$

Because of the limited flexion range of the spine, the turning radius is constrained by the mechanical design. It should be noted that, during the spine lateral flexion turning, the leg stride length does not need to be modulated to achieve the turning behavior.

The leg-based and spine-based turning strategies have several benefits and drawbacks, as outlined in table S6. We introduced a hybrid turning strategy called mix-based turning, which combines the advantages of both leg-based and spine-based turning strategies. The mix-based turning strategy applies the following logic to combine turning strategies. For large turning radii, the mix-based turning prioritizes the faster spine-based turning. For small turning radii, the mix-based strategy applies additional leg-based turning. The turn radius is computed in the same way as leg-based turning, except that d_{diag} is also conditioned on the spine vertebral angle θ_v.

Control architecture

The control architecture integrates the kinematic models of the leg and spine into a cohesive system to achieve spine-based quadruped locomotion. The control architecture consists of four components: the high-level controller (see the Supplementary Materials), the motion module, the trajectory generator, and the gait knowledge base. The control architecture relies on synchronized communication between all modules to achieve successful control. In Fig. 8A, we show the information flow, where $\boldsymbol{k}_g$ is the gait parameter vector, $\boldsymbol{s}_{legs}$ is the leg state vector, $\boldsymbol{v}_{legs}$ is the leg velocity vector, $\boldsymbol{tr}_{foot}$ is the foot-end trajectory, v_m is the target velocity of the robot, α_{turn} is the turning rate, and $\boldsymbol{g}_d$ is the gait mode. The controller has two external interfaces (i) to an upstream input device that sends control messages $\boldsymbol{c}_m$ and (ii) to the simulation or NeRmo with control signal $\boldsymbol{q}_{m,d}$ and sensor messages $\boldsymbol{q}_{m,s}$.

The motion module implements the kinematic models of NeRmo's legs and spine. In particular, the motion module must ensure that the leg and spine are synchronized for all behaviors. Figure 8B depicts the motion module and the interdependencies between the low-level leg and spine controllers. In addition to the internal communication between the leg and spine controller, the motion module's main interface sends the servo control values $\boldsymbol{q}_{m,d}$ to the downstream device. It also interfaces with the gait knowledge base to retrieve relevant gait parameters $\boldsymbol{k}_g$ and the trajectory generator to obtain the foot-end trajectories $\boldsymbol{tr}_{foot}$. The external and internal control signals of the motion module are outlined in table S12. Note that the parameter $\boldsymbol{q}_{m,d} = (\boldsymbol{q}_{legs}, q_{sp})$ is depicted in Fig. 8A. Detailed explanations of the spine controller, leg controller, and the trajectory generator are given in the Supplementary Materials.

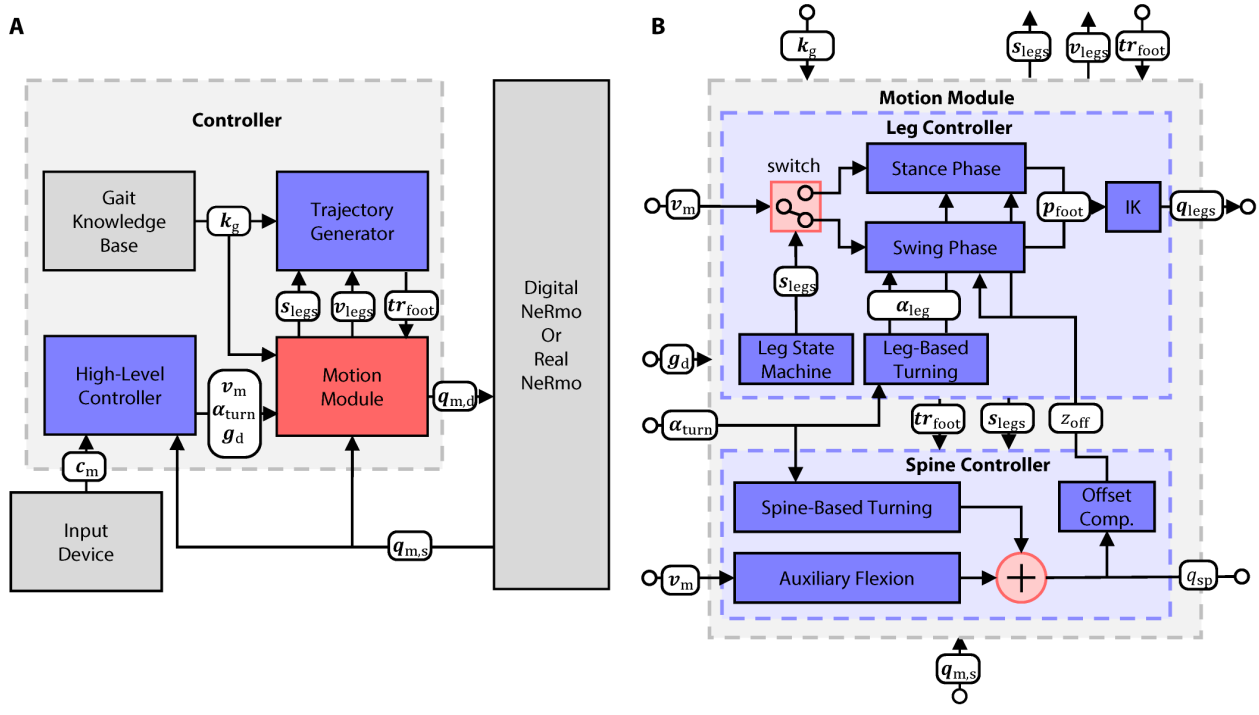

Fig. 8. Overview of the control architecture. (**A**) The controller consists of four core modules: the high-level controller, the motion module, the trajectory generator (fig. S11), and the gait knowledge base. The controller interfaces with two external components. The first component is any upstream input device that sends a control message $\boldsymbol{c}_m$. The second component sends servo controls $\boldsymbol{q}_{m,d}$ and receives sensor values $\boldsymbol{q}_{m,s}$ from the digital or real NeRmo. (**B**) Architecture of the motion module, which computes the low-level control commands for the servo motors.

Supplementary Materials

This PDF file includes:
Supplementary Methods
Figs. S1 to S18
Tables S1 to S12
References (*49*–*63*)

Other Supplementary Material for this manuscript includes the following:
Movies S1 to S12

REFERENCES AND NOTES

1. M. H. Dickinson, C. T. Farley, R. J. Full, M. A. R. Koehl, R. Kram, S. Lehman, How animals move: An integrative view. *Science* **288**, 100–106 (2000).
2. S. Coros, A. Karpathy, B. Jones, L. Reveret, M. Van De Panne, Locomotion skills for simulated quadrupeds. *ACM Trans. Graph.* **30**, 1–12 (2011).
3. J. Dvorák, M. Panjabi, D. Chang, R. Theiler, D. Grob, Functional radiographic diagnosis of the lumbar spine: Flexion–extension and lateral bending. *Spine* **16**, 562–571 (1991).
4. R. Sullivan, *Rats: Observations on the History and Habitat of the City's Most Unwanted Inhabitants* (Bloomsbury USA, 2004).
5. Z. Zhang, J. Yang, H. Yu, Effect of flexible back on energy absorption during landing in cats: A biomechanical investigation. *J. Bionic Eng.* **11**, 506–516 (2014).
6. C. J. Pycock, Turning behaviour in animals, in *Commentaries in the Neurosciences*, A. D. Smith, R. Llinas, P.G. Kostyuk, Eds., (Elsevier, 1980), pp. 461–512.
7. Verge, Boston Dynamics Spot (2019) [online; accessed 9 November 2021].
8. G. Bledt, M. J. Powell, B. Katz, J. D. Carlo, P. M. Wensing, S. Kim, MIT cheetah 3: Design and control of a robust, dynamic quadruped robot, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018) pp. 2245–2252.
9. M. Hutter, C. Gehring, D. Jud, A. Lauber, C. Dario Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, M. Hoepflinger, ANYmal—A highly mobile and dynamic quadrupedal robot, in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2016), pp. 38–44.
10. J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, M. Hutter, Learning agile and dynamic motor skills for legged robots. *Sci. Robot.* **4**, eaau5872 (2019).
11. T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci. Robot.* **7**, eabk2822 (2022).
12. Y. Yesilevskiy, W. Yang, C. D. Remy, Spine morphology and energetics: How principles from nature apply to robotics. *Bioinspir. Biomim.* **13**, 036002 (2018).
13. M. Khoramshahi, A. Spröwitz, A. Tuleu, M. N. Ahmadabadi, A. J. Ijspeert, Benefits of an active spine supported bounding locomotion with a small compliant quadruped robot, in *2013 IEEE International Conference on Robotics and Automation* (IEEE, 2013), pp. 3329–3334.
14. S. Bhattacharya, A. Singla, Abhimanyu, D. Dholakiya, S. Bhatnagar, B. Amrutur, A. Ghosal, S. Kolathaya, 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)* (IEEE, 2019), pp. 1–6.
15. C. Fisher, S. Shield, A. Patel, The effect of spine morphology on rapid acceleration in quadruped robots, in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2017), pp. 2121–2127.
16. J. Or, Humanoids grow a spine: The effect of lateral spinal motion on the mechanical energy efficiency. *IEEE Robot. Autom. Mag.* **20**, 71–81 (2013).
17. Q. Zhao, H. Sumioka, R. Pfeifer, The effect of morphology on the spinal engine driven locomotion in a quadruped robot, in *The 5th International Symposium on Adaptive Motion of Animals and Machines* (Springer, 2011), pp. 51–52.
18. A. P. Sabelhaus, Lara Janse van Vuuren, A. Joshi, E. Zhu, H. J. Garnier, K. A. Sover, J. Navarro, A. K. Agogino, A. M. Agogino, Design, simulation, and testing of a flexible actuated spine for quadruped robots, arXiv preprint arXiv:1804.06527 (2018).
19. Verge, Boston dynamic cheetah (2012). [Online; accessed Nov. 9, 2021].
20. P. Eckert, A. Spröwitz, H. Witte, A. J. Ijspeert, Comparing the effect of different spine and leg designs for a small bounding quadruped robot, in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), pp. 3128–3133.
21. K. Weinmeister, P. Eckert, H. Witte, A. J. Ijspeert, Cheetah-cub-s: Steering of a quadruped robot using trunk motion, in *2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (IEEE, 2015), pp. 1–6.
22. S. Pouya, M. Khodabakhsh, A. Spröwitz, A. Ijspeert, Spinal joint compliance and actuation in a simulated bounding quadruped robot. *Auton. Robots* **41**, 437–452 (2017).
23. H. Ishii, Y. Masuda, S. Miyagishima, S. Fumino, A. Takanishi, C. Laschi, B. Mazzolai, V. Mattoli, P. Dario, Design and development of biomimetic quadruped robot for behavior studies of rats and mice, in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (IEEE, 2009), pp. 7192–7195.
24. Q. Shi, C. Li, K. Li, Q. Huang, H. Ishii, A. Takanishi, T. Fukuda, A modified robotic rat to study rat-like pitch and yaw movements. *IEEE/ASME Trans. Mechatron.* **23**, 2448–2458 (2018).
25. S. Wang, Q. Shi, J. Gao, Y. Wang, F. Meng, C. Li, Q. Huang, T. Fukuda, Design and control of a miniature quadruped rat-inspired robot, in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (IEEE, 2019), pp. 346–351.
26. Q. Shi, J. Gao, S. Wang, X. Quan, G. Jia, Q. Huang, T. Fukuda, Development of a small-sized quadruped robotic rat capable of multimodal motions. *IEEE Trans. Robot.* **38**, 3027–3043 (2022).
27. Q. Zhao, K. Nakajima, H. Sumioka, X. Yu, R. Pfeifer, Embodiment enables the spinal engine in quadruped robot locomotion, in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2012), pp. 2449–2456.
28. B. Chong, Y. O. Aydin, C. Gong, G. Sartoretti, Y. Wu, J. M. Rieser, H. Xing, P. E. Schiebel, J. W. Rankin, K. B. Michel, A. Nicieza, J. R. Hutchinson, D. I. Goldman, H. Choset, Coordination of lateral body bending and leg movements for sprawled posture quadrupedal locomotion. *Int. J. Rob. Res.* **40**, 747–763 (2021).
29. W. Haomachai, D. Shao, W. Wang, A. Ji, Z. Dai, P. Manoonpong, Lateral undulation of the bendable body of a gecko-inspired robot for energy-efficient inclined surface climbing. *IEEE Robot. Autom. Lett.* **6**, 7917–7924 (2021).
30. A. J. Ijspeert, A. Crespi, D. Ryczko, J.-M. Cabelguen, From swimming to walking with a salamander robot driven by a spinal cord model. *Science* **315**, 1416–1420 (2007).
31. A. Crespi, K. Karakasiliotis, A. Guignard, A. J. Ijspeert, Salamandra robotica ii: An amphibious robot to study salamander-like swimming and walking gaits. *IEEE Trans. Robot.* **29**, 308–320 (2013).
32. A. J. Ijspeert, Biorobotics: Using robots to emulate and investigate agile locomotion. *Science* **346**, 196–203 (2014).
33. Y. Tang, Y. Chi, J. Sun, T.-H. Huang, O. H. Maghsoudi, A. Spence, J. Zhao, H. Su, J. Yin, Leveraging elastic instabilities for amplified performance: Spine-inspired high-speed and high-force soft robots. *Sci. Adv.* **6**, eaaz6912 (2020).
34. Q. Zhao, B. Ellenberger, H. Sumioka, T. Sandy, R. Pfeifer, The effect of spine actuation and stiffness on a pneumatically-driven quadruped robot for cheetah-like locomotion, in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (IEEE, 2013), pp. 1807–1812.
35. J. Lei, H. Yu, T. Wang, Dynamic bending of bionic flexible body driven by pneumatic artificial muscles(PAMs) for spinning gait of quadruped robot. *Chin. J. Mech. Eng.* **29**, 11–20 (2016).
36. K. Narioka, A. Rosendo, A. Sproewitz, K. Hosoda, Development of a minimalistic pneumatic quadruped robot for fast locomotion, in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (IEEE, 2012), pp. 307–311.
37. K. Karakasiliotis, R. Thandiackal, K. Melo, T. Horvat, N. K. Mahabadi, S. Tsitkov, J. M. Cabelguen, A. J. Ijspeert, From cineradiography to biorobots: An approach for designing robots to emulate and study animal locomotion. *J. R. Soc. Interface* **13**, 20151089 (2016).
38. P. Lucas, S. Oota, J. Conradt, A. Knoll, Development of the neurorobotic mouse, in *2019 IEEE International Conference on Cyborg and Bionic Systems (CBS)* (IEEE, 2019), pp. 299–304.
39. M. Hildebrand, Symmetrical gaits of horses: Gaits can be expressed numerically and analyzed graphically to reveal their nature and relationships. *Science* **150**, 701–708 (1965).
40. R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, M. P. Deisenroth, Bayesian gait optimization for bipedal locomotion, in *Learning and Intelligent Optimization: 8th International Conference* (Springer, 2014), pp. 274–290.
41. D. Lizotte, T. Wang, M. Bowling, D. Schuurmans, Automatic gait optimization with gaussian process regression, in *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007* (Morgan Kaufmann Publishers Inc., 2007), pp. 944–949.
42. M. Tesch, J. Schneider, H. Choset, Using response surfaces and expected improvement to optimize snake robot gait parameters, in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2011), pp. 1069–1074.
43. K. Zhai, C. Li, A. Rosendo, Scaffolded, Learning of in-place trotting gait for a quadruped robot with Bayesian optimization, in *Intelligent Autonomous Systems 16: Proceedings of the 16th International Conference IAS-16* (Springer, 2022), pp. 365–373.
44. J. Zhu, S. Li, Z. Wang, A. Rosendo, Bayesian optimization of a quadruped robot during 3-dimensional locomotion, in *Biomimetic and Biohybrid Systems: 8th International Conference, Living Machines 2019* (Springer, 2019), pp. 295–306.
45. Y. Zhong, R. Wang, H. Feng, Y. Chen, Analysis and research of quadruped robot's legs: A comprehensive review. *Int. J. Adv. Robot. Syst.* **16**, 172988141984414 (2019).
46. J. Larsen, "Locomotion through morphosis," thesis, University of Southern Denmark, Odense, Denmark (2013).
47. D. Kuehn, A. Dettmann, F. Kirchner, F. Kirchner, Analysis of using an active artificial spine in a quadruped robot, in *2018 4th International Conference on Control, Automation and Robotics (ICCAR)* (IEEE, 2018), pp. 37–42.
48. C. Bellardita, O. Kiehn, Phenotypic characterization of speed-associated gait changes in mice reveals modular organization of locomotor networks. *Curr. Biol.* **25**, 1426–1436 (2015).
49. Q. Q. Liang, G. P. Steven, A performance-based optimization method for topology design of continuum structures with mean compliance constraints. *Comput. Methods Appl. Mech. Eng.* **191**, 1471–1489 (2002).

50. M. Liu, X. Zhang, S. Fatikow, Design of flexure hinges based on stress-constrained topology optimization. *Proc. Inst. Mech. Eng. C J. Mech. Eng. Sci.* **231**, 4635–4645 (2017).
51. J. Pinskier, B. Shirinzadeh, Topology optimization of leaf flexures for stiffness ratio maximization in compliant mechanisms, in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (IEEE, 2018), pp. 407–412.
52. S. Koppen, M. Langelaar, F. van Keulen, A simple and versatile topology optimization formulation for flexure synthesis. *Mech. Mach. Theory* **172**, 104743 (2022).
53. S. Chen, F. Chen, Z. Cao, Y. Wang, Y. Miao, G. Gu, X. Zhu, Topology optimization of skeleton-reinforced soft pneumatic actuators for desired motions. *IEEE/ASME Trans. Mechatron.* **26**, 1745–1753 (2021).
54. Pa 1101-polyamide 11; https://store.eos.info/products/pa-1101-polyamide-11 [accessed 25 April 2023].
55. Pa 2200-polyamide 12; https://store.eos.info/products/pa-2200-polyamide-12 [accessed 25 April 2023].
56. Pa 3200 gf–polyamide 12; https://store.eos.info/products/pa-3200-gf-polyamide-12 [accessed 25 April 2023].
57. L. Rhode-Barbarigos, C. Schulin, N. B. H. Ali, R. Motro, I. F. Smith, Mechanism-based approach for the deployment of a tensegrity-ring module. *J. Struct. Eng.* **138**, 539–548 (2012).
58. Verge. AIRO IDLab Ghent University—Robot hardware.
59. D. Zappetti, R. Arandes, E. Ajanic, D. Floreano, Variable-stiffness tensegrity spine. *Smart Mater Struct.* **29**, 075013 (2020).
60. S. M. Levin, The tensegrity-truss as a model for spine mechanics: Biotensegrity. *J. Mech. Med. Biol.* **02**, 375–388 (2002).
61. S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, S. Kim, Design principles for highly efficient quadrupeds and implementation on the MIT cheetah robot, in *2013 IEEE International Conference on Robotics and Automation* (IEEE, 2013), pp. 3307–3312.
62. E. Todorov, T. Erez, Y. Tassa, MuJoCo: A physics engine for model-based control, in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2012), pp. 5026–5033.
63. L. De Novellis, A. Sorniotti, P. Gruber, L. Shead, V. Ivanov, K. Hoepping, Torque vectoring for electric vehicles with individually controlled motors: State-of-the-art and future developments. *World Electr. Veh. J.* **5**, 617–628 (2012).

Acknowledgments: We thank Y. Chen for the valuable suggestion on the finite element analysis simulation of the spine flexion. We thank Y. Xiao for the valuable discussion on the development of the grid search controller and the Bayesian optimization controller. F.W. has been affiliated with the Chair of Robotics, Artificial Intelligence and Real-Time Systems, TUM School of Computation, Information and Technology, Technical University of Munich while the research was conducted and is now primarily affiliated with the Machine Intelligence Lab, Engineering Department, University of Technology Nuremberg. **Funding:** This study was supported by the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement no. 785907 (Human Brain Project SGA2) and Specific Grant Agreement no. 945539 (Human Brain Project SGA3) and Pazhou Lab PZL2021KF0020, Guangzhou, China. **Author contributions:** Conceptualization: Z.B., A.R., F.W., Y.H., P.L., F.O.M., K.H., and A.K. Methodology: Z.B., A.R., F.W., Y.H., P.L., K.H., and A.K. Investigation: Z.B., A.R., F.W., Y.H., and P.L. Funding acquisition: A.K. and K.H. Writing—original draft: Z.B., A.R., F.W., Y.H., F.O.M., K.H., and A.K. Writing—review and editing: Z.B., A.R., F.W., Y.H., P.L., F.O.M., K.H., and A.K. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data are available in the main text, the Supplementary Materials, or at the following link: https://doi.org/10.5281/zenodo.10064687.

Submitted 17 January 2023
Accepted 7 November 2023
Published 6 December 2023
10.1126/scirobotics.adg7165

Science Robotics

Lateral flexion of a compliant spine improves motor performance in a bioinspired mouse robot

Zhenshan Bing, Alex Rohregger, Florian Walter, Yuhong Huang, Peer Lucas, Fabrice O. Morin, Kai Huang, and Alois Knoll

Sci. Robot. **8** (85), eadg7165. DOI: 10.1126/scirobotics.adg7165

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.