

Spined Torso Renders Advanced Mobility for Quadrupedal Locomotion

Jichao Wang, Jinyu Cheng, Jiangtao Hu, Wei Gao and Shiwu Zhang

Abstract—Animals possessing spinal columns often exhibit exceptional agility for highly dynamic locomotion. The spine grants the trunk with increased degrees of freedom, thereby endowing diverse postures. This paper presents the development of a robot STRAY for quadrupedal locomotion, featuring a four-degree-of-freedom spine design. Using trajectory based reinforcement learning techniques, STRAY is able to trot and bound dynamically using its spine. Simulation results reveal the positive roles of spinal movement, such as twisting, extension, retraction and rotation, in helping STRAY realize efficient locomotion. Preliminary results from experiments demonstrate that STRAY can achieve a trotting gait of approximately 0.6 m/s and a bounding gait of 0.7 m/s, with desired velocities of 0.8 m/s and 1.0 m/s, respectively. The results also indicate that reinforcement learning is a feasible way to investigate how the spine should be used in dynamic quadrupedal locomotion and achieve more possibilities in the future.

I. INTRODUCTION

Quadrupeds in the natural world, such as cheetahs, horses, dogs, cats, *etc.*, exhibit remarkable dynamic locomotion abilities. They have inspired researchers to replicate these abilities onto various types of quadrupedal robots [1], [2], [3], [4]. Although these robots' mobility has been impressively improved through continuous efforts in recent years [5], they still lag behind their natural world's counterparts and only present limited dynamic performance. The gap lies in both system design and locomotion control, where finite control authority to the system structure cannot afford muscle-like actuation and limited onboard computation resources has impeded highly dynamic locomotion control.

Consequently, most research on quadrupedal robots has been favoring the single-rigid-body design and control, which appears to have reached a plateau due to its limited degrees of freedom. Actually, vertebral column has been commonly seen in quadrupeds to enable diverse body postures during dynamic locomotion [6]. Taking the jumping ability of cheetahs for instance, their hind legs can utilize the flexibility of their vertebral column to generate a greater range of motion, performing explosively powerful leaps [7]. In contrast, current single-rigid-body quadrupedal robots are unable to replicate this feat.

This work was supported by the National Natural Science Foundation of China under 62103395, U21A20119 and U22B2040. (Jichao Wang and Jinyu Cheng are co-first authors.) (Corresponding authors: Wei Gao; Shiwu Zhang.

Jichao Wang, Jinyu Cheng, Wei Gao and Shiwu Zhang are with the Department of Precision Machinery and Precision Instrumentation, College of Engineering Science, University of Science and Technology of China, Hefei, Anhui 230026, China. (e-mail: weigao@ustc.edu.cn; swzhang@ustc.edu.cn)

Jiangtao Hu is with the Institute of Advanced Technology, University of Science and Technology of China, Hefei, Anhui 230088, China.

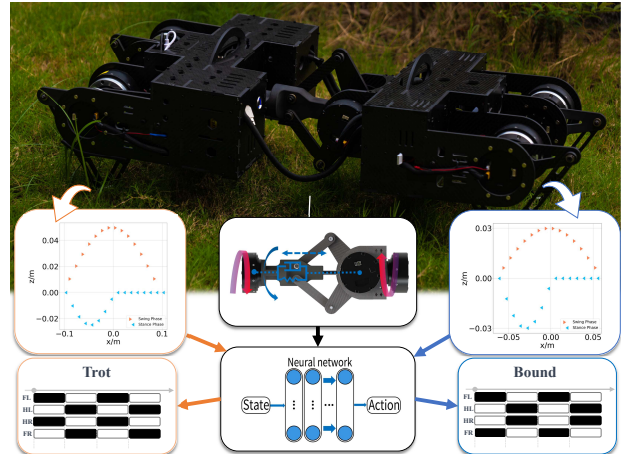


Fig. 1: Overview: the quadrupedal robot STRAY with a four-degree-of-freedom spine and the hybrid controller combining foot reference trajectories and spine control policies.

It is encouraging that researchers have started to consider incorporating actively controlled spines into quadrupedal robots. The robot Inu from University of Pennsylvania possesses a parallel elastic-actuated spine, allowing the spinal section to tilt up and down in the sagittal plane. By employing a reduced-order model for the core-actuation, the complexity of its dynamic analysis can be potentially reduced [8]. Experimental results have also shown that judiciously using spinal bending is advantageous for quadrupedal robots to cross over higher obstacles [9]. Their recent study deploys a twisting spine to a quadrupedal robot and realizes highly dynamic wall jumping behavior in simulation through model-based optimization [10]. These investigations into unidirectional spine functionality in quadrupedal locomotion helps establish preliminary understanding of such mechanism. To investigate the effect of multi-degree-of-freedom spines, researchers from India have developed a quadrupedal robot called Stoch 2 with a two degrees of freedom (DOFs) spine, which shows that active spine can increase stride length during bounding [11]. Also recently, another quadrupedal robot, Yat-sen Lion, incorporates a three-degree-of-freedom spine to realize fore-aft body tilting in both sagittal and lateral planes simultaneously, thus achieving more diverse body postures during locomotion [12]. The major control challenges of quadrupedal robots with multi-degree-of-freedom spines come from the variation of its Center of Mass (COM) and moment of inertia during locomotion due to spine-caused body reconfiguration. Yat-sen Lion integrates the calculation of variable COM and moment of inertia into

the Model Predictive Control (MPC) framework, showing a feasible way to realize dynamic locomotion. However, knowledge of how to judiciously utilize the spine is not provided.

To further explore the role of spines in quadrupedal locomotion, this paper proposes a four-degree-of-freedom spine designed in the form of a symmetric five-bar mechanism, as shown in Figure 1. The two rolling DOFs at both ends of the spine allow it to rotate about the fore-aft body axis and thus the other two DOFs within the five-bar mechanism can realize the extension, retraction and rotation of the planar spine in any selected plane. The quadrupedal robot incorporating this spine is thus named STRAY (Spined Torso Renders Advanced mobilitY). To keep a consistent style, the leg design of STRAY inherits the symmetric five-bar design from the quadrupedal robot Minitaur [4], with two sagittal-plane degrees of freedom per leg. As a result, the robot has 12 DOFs in total, which is the same as that of most single-rigid-body quadrupedal robots. It is worth noting that this version of STRAY purposely reallocates the four hip DOFs to form an active spine, in hope of studying the effect of utilizing these DOFs in this different manner and making a comparison.

On the other hand, as mentioned, the multi-DOF spine poses a great control challenge to using the prevailing MPC control framework for single-rigid-body quadrupedal robots [13]. Even though the MPC framework can be adapted to variable COM and moment of inertia for computation and control convenience, it still does not teach the robot how to judiciously use a spine and thus limits the benefit provided by it. In recent years, reinforcement learning (RL) based methods for robotic control has emerged and become popular due to its thorough exploration of the state-action space during training and the robust optimal controllers, or policy networks, obtained from training. The most challenging portion of implementing RL based control method is often the reward function design and the sim-to-real transfer, where the former requires judicious selection and the latter requires tremendous effort and patience. Therefore, to explore how to dynamically control the four-DOF spine on STRAY, this work applies RL to train a policy network. The obtained policy successfully helps STRAY achieve dynamic bounding gaits and reveals how the multi-DOF spine should be actively controlled to achieve advanced mobility for quadrupedal robots.

The remainder of the paper first introduces the design details of STRAY in Section II. Following that, Section III presents the proposed control method to help STRAY realize dynamic gaits. The corresponding simulation and experimental results are then discussed in Section IV. Finally, Section V concludes the paper.

II. PHYSICAL PLATFORM

A. Mechanical design

STRAY weighs approximately 16.35 kg, with a nominal dimension of 700 mm in length, 411 mm in width and 284 mm in height, as shown in Figure 2. As mentioned,

it features a spinal architecture that connects the front and the rear body bases. The bases are constructed using in total 26 pieces of 5 mm thick carbon fiber sheets, which are assembled together using mortise-and-tenon joints and secured in place with M2.5 screws.

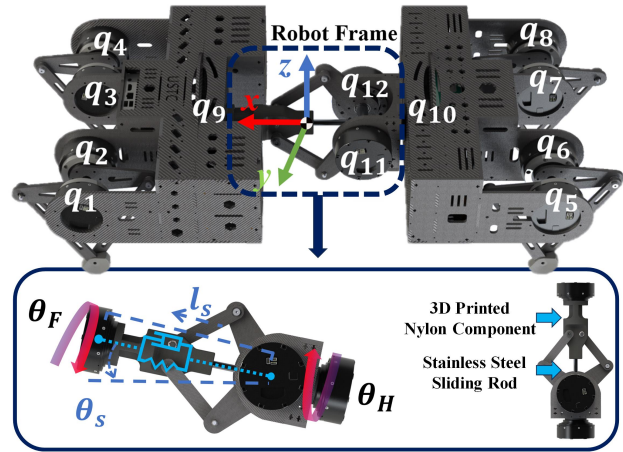


Fig. 2: Details of STRAY's mechanical design with the notation of the 12 DOFs.

For each leg, the linkages of the five-bar mechanism are made of well-cut carbon fiber sheets, with the length of the shorter linkages being 10 cm and that of the longer ones being 20 cm. Two Unitree A1 brushless DC motors with a gear reduction ratio of 9.1 : 1 are placed coaxially at the hip for actuation. Each motor weighs approximately 620 g. The motors are operated under 24 V voltage input and can deliver a peak torque of 32.5 N m. At the distal end of the leg, a 3D printed foot wrapped with rubber is attached for reducing the impact with the ground.

Similarly, the spine is also made of well-cut carbon fiber sheets, as well as a 3D printed nylon component and a stainless steel sliding rod, as shown in the insert in Figure 2. The four major linkages of the spine all have the same length of 10 cm, and the same Unitree A1 motors are used for actuating the four DOFs. Two motors are placed at both ends of the spine to collaboratively rotate it along the fore-aft axis, such that the extension, retraction and rotation of the spine by the other two motors can perform in any selected plane. Note that the end with three motors is connected to the rear base, while the end with only one motor is connected to the front base. This symmetric five-bar structural design can potentially help even up the torque distribution between the two planar motors in the spine, and thus maximize the spine's axial force output when needed [14]. To help stabilize axial movement of the spine, *i.e.* extension and retraction, the sliding rod is used to maintain the spine's orientation. The sliding rod measures 153 mm in length and 10 mm in diameter, enabling the spine to realize axial movement of at least 10 cm.

It is worth noting that when implementing the actual control to these five-bar mechanisms, the lengths and angles of the virtual legs and the virtual spine are manipulated rather

than the direct motor angles for computational efficiency and algorithmic simplicity.

B. Electronic design

To control STRAY to achieve desired dynamic gaits, an Intel NUC 12 Pro running Ubuntu 20.04 is equipped as the upper-level onboard computer. It communicates with two STM32F767 based microcontrollers via USB cables. These two microcontrollers serve as the lower-level controllers to communicate with the 12 motors through Controller Area Network (CAN) bus. Additionally, the system utilizes a WheelTec N200 Inertia Measurement Unit (IMU) to obtain the states of the front base, including linear accelerations and angular velocities. The communication between the motors, the lower-level microcontroller and the upper-level computer all operate at a frequency of 1000 Hz, while the IMU data acquisition runs at 200 Hz, as shown in Figure 3.

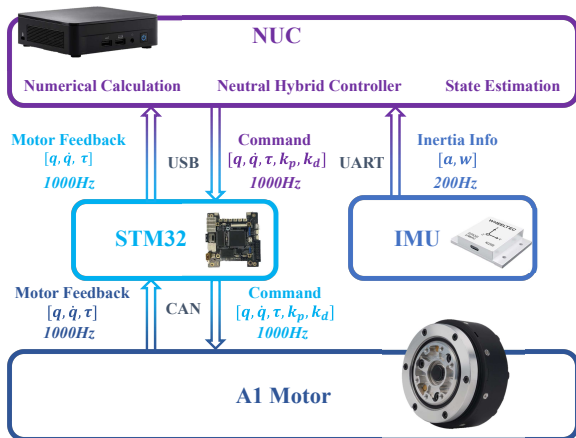


Fig. 3: Electronic design details of STRAY.

III. HYBRID CONTROLLER

A. Reference foot trajectories

To simplify the RL process of the spine control, this paper assigns reference trajectories for the robot’s feet position in the feet task space, as shown in Figures 4(a) and 4(b) for trotting and bounding gaits, respectively. For the trotting gait foot trajectory, the swing phase represented by orange right triangles is a sinusoidal curve, while the stance phase represented by blue left triangles is consisted of a horizontal straight line and a small downward sinusoidal trajectory. For the bounding gait foot trajectory, the swing phase follows the same sinusoidal trajectory, while the stance phase trajectory consists of two parts, with the first half being a straight line parallel to the x-axis, and the second half being a downward sinusoidal trajectory with the same amplitude as the swing phase that allows the robot to push against the ground to jump upward. Besides, the length of the foot trajectory along the x-axis varies according to the speed, described as:

$$l_x = \frac{T \cdot v_{des}}{2} \quad (1)$$

where T represents the gait period and v_{des} represents the desired speed. The step height, which is the amplitude of

the sine trajectory, is set to 5 cm for trotting gaits and 3 cm for bounding gaits.

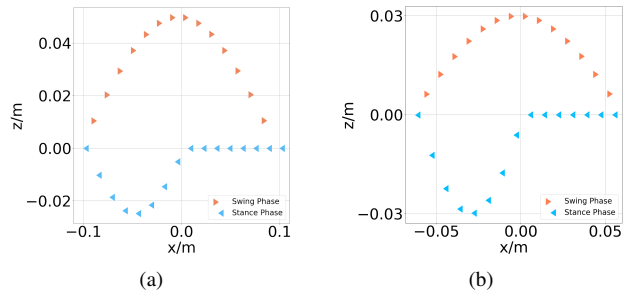


Fig. 4: The nominal foot trajectories. (a) The foot trajectory of the 0.8 m/s trotting gait. (b) The foot trajectory of the 1.0 m/s bounding gait.

B. RL based hybrid controller

To enable STRAY to locomote using its spine dynamically, a model-free RL approach is utilized to add a learning based controller on top of the aforementioned foot trajectory based leg controller. The resulting hybrid controller can be expressed as

$$\mathbf{A}(t) = \begin{bmatrix} \mathbf{A}_{leg}(t) \\ \mathbf{A}_{spine}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{a}(t) \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\rho} \\ \mathbf{1} \end{bmatrix}^T \boldsymbol{\pi}(\mathbf{x}_t) \quad (2)$$

where $\boldsymbol{\pi}(\mathbf{x}_t)$ represents the learned policy that can generate the feedback action given current system states \mathbf{x}_t , $\mathbf{a}(t)$ represents the desired joint angles for following the nominal time-based foot trajectories, $\mathbf{A}(t)$ represents the resultant action for all the joints and the subscripts *leg* and *spine* indicate the specific portion of action for the four legs and the spine, respectively. The vector $\boldsymbol{\rho}$ is a control weight between 0 and 1 to adjust the level of influence from the learned policy on the foot trajectories. It can be seen that the hybrid control is only applied on the leg motors, while the actions of the spine motors are completely learned. This approach can obtain a balance between exploration and exploitation, allowing for comprehensive exploration of the problem space while exploiting existing knowledge and data.

C. Training details

1) *State-action space*: The state-action space in this work is defined as follows:

State: The state of the robot \mathbf{x}_t is a 30-dimensional vector, including 3-dimensional front-base linear acceleration, 3-dimensional front-base angular velocity, 12-dimensional joint positions \mathbf{q} , and 12-dimensional joint velocities $\dot{\mathbf{q}}$. Note that these are all the states that can be directly obtained from the IMU and the motor encoders on STRAY, where the angular velocity is obtained from IMU with built-in algorithms.

Action: The action of the robot $\mathbf{A}(t)$ is a 12-dimensional vector, including the virtual length and angle of the four legs, l_L and θ_L , the rolling angles of the front and the rear bases with respect to the spine, θ_F and θ_H , and the virtual length and angle of the spine, l_S and θ_S . The actions

$\mathbf{A}(t)$ need to be converted into desired motor positions using inverse kinematics, and then into desired motor torques using a proportional-derivative (PD) controller defined as

$$\boldsymbol{\tau} = K_p(\hat{\mathbf{q}} - \mathbf{q}) + K_d(\hat{\dot{\mathbf{q}}} - \dot{\mathbf{q}}) \quad (3)$$

where $\hat{\mathbf{q}}$ and $\hat{\dot{\mathbf{q}}}$ represents the desired angular positions and velocities of the 12 motors. Note that the desired velocity $\hat{\dot{\mathbf{q}}}$ is set to $\mathbf{0}$.

2) *Reward function*: In the learning process, five categories of reward are given to the robot to encourage it to move forward dynamically using its spine and realize steady locomotion. The five corresponding terms are mathematically expressed as follows:

$$\begin{cases} r_v = \exp(-\|\mathbf{v} - \mathbf{v}_{des}\|^2/(2\sigma_1)) \\ \sigma_1 = 0.05 \cdot \mathbf{v}_{des} \\ r_a = \exp(-\|\mathbf{a}\|^2/(2\sigma_2)) \\ \sigma_2 = 0.1 \\ r_y = |\theta_{yaw}| \\ r_e = \exp(-\sum_{i=1}^{12} \max\{0, \tau_i^2 \dot{q}_i^2\}/(2\sigma_3)) \\ \sigma_3 = 25 \\ r_s = \sum_{i=1}^8 \|\frac{\Delta\pi_i}{\bar{\pi}_i}\|^2 + 8 \cdot \sum_{i=9}^{12} \|\frac{\Delta\pi_i}{\bar{\pi}_i}\|^2 \\ \Delta\pi_i = \pi_i(\mathbf{x}_t) - \pi_i(\mathbf{x}_{t-1}) \end{cases} \quad (4)$$

The first term r_v rewards the robot for approaching the desired speed and move forward in a consistent manner, which is critical for successful locomotion. In the expression, \mathbf{v} indicates the robot’s current velocity, \mathbf{v}_{des} is the desired velocity, and σ_1 is a hyperparameter that controls the spread of the reward distribution. The second term r_a rewards the robot for maintaining a constant forward speed, where \mathbf{a} is the acceleration of the front base. The third term r_y penalizes the robot’s body rotation around the z -axis, especially caused by the spine. The fourth term r_e penalizes the robot for using excessive motor power. The last term r_s helps smooth up the policy by avoiding too much variation between consecutive actions, where the notation $\pi_i(\mathbf{x}_t)$ refers to the portion of policy for the i -th joint action, and $\bar{\pi}_i$ refers to the range of that generated action. The first eight items are actions for the legs, while the last four items are actions for the spine.

In summary, the overall reward function can be expressed as:

$$\mathbf{r} = \mathbf{w}^T \mathbf{r} \quad (5)$$

where

$$\begin{cases} \mathbf{w} = [40, 5, -10, 10, -2]^T \\ \mathbf{r} = [r_v, r_a, r_y, r_e, r_s]^T \end{cases} \quad (6)$$

3) *Training process*: To train the RL based policy network $\pi(\mathbf{x}_t)$ and simulate the proposed hybrid control approach using $\mathbf{A}(t)$, the robust physics engine MuJoCo [15] is utilized. MuJoCo is designed to simulate the dynamics of complex multi-body systems and also features an interface for implementing RL algorithms as well as the learned policies, making it an excellent fit for the purpose of

this paper. Specifically, the default Multilayer Perceptron (MLP) from Stable Baselines3 [16] is adopted for the policy network. The built-in Proximal Policy Optimization (PPO) algorithm [17] is used to train the desired policy network, which takes system states as input and generates desired action for the spine and desired adjustment action for the legs as output. During the training process, STRAY’s joint motors are operated at a frequency of 1000 Hz, and the desired action for those joint motors are generated by the learned policy at a frequency of 50 Hz.

4) *Sim to real transfer*: RL is a powerful approach to teach robots to dynamically interact with the environment through high-fidelity simulators. However, there is often gaps between the simulation environment and the real-world scenarios, which can affect the performance of the learned policies on the actual robots. To overcome this issue, parameter randomization in the simulation environment is conducted to better approximate real-world conditions, as well as adding 20% Gaussian noise to the observation. This randomization helps improve the generalization of learned policies and make the robot more robust to unforeseen situations. The specific ranges for randomized parameters and sensor noises are listed in Table I, while the hyperparameter values for the training process can be found in Table II.

TABLE I: Randomization ranges for physical parameters and sensor noises.

| Name | Lower bound | Upper bound |
|---------------|-------------|-------------|
| $k_{p,leg}$ | 30 | 50 |
| $k_{d,leg}$ | 0.6 | 1.0 |
| $k_{p,spine}$ | 100 | 150 |
| $k_{d,spine}$ | 2 | 4 |
| Mass | 50% | 150% |
| Inertia | 50% | 150% |
| CoM | 50% | 150% |
| Friction | 0.6 | 1.0 |

TABLE II: PPO hyperparameter values.

| Parameter | Value |
|-----------------|---|
| Optimizer | Adam |
| Learning rate | $3e-4 \xrightarrow{\text{progress}} 1e-4$ |
| Clip range | $0.2 \xrightarrow{\text{progress}} 0$ |
| GAE parameter | 0.9 |
| Mini-batch size | 256 |
| Discount factor | 0.99 |

IV. RESULTS AND DISCUSSIONS

To gain a better understanding of the spine’s role in dynamic quadrupedal locomotion, a series of experiments are conducted in both simulation and real world, the results of which are presented and discussed in the following sub sections.

A. Experimental setup

During the experiments, the control frequency of the robot's actuators was set to 1000 Hz, while the policy's output frequency was set to 50 Hz. The joint motors were set to torque control mode, and the torques were calculated using equation (3) with the corresponding proportional-derivative gains fixed as $K_p = 50$ and $K_d = 1$. The RL based hybrid controller is trained and implemented for both the trotting gait and the bounding gait. During training, the ranges of STRAY's action space is defined as follows: the virtual leg angle θ_L is confined within the range of $(-0.52, 0.52)$ radians, the virtual leg length l_L is confined within the range of $(-0.05, 0.05)$ meters, the front and rear base rolling angles θ_F and θ_H are confined within the range of $(-0.1, 0.1)$ radians, the spine angle θ_S is confined with the range of $(0, 0.25)$ radians, and the spine length variation Δl_S is confined within the range of $(-0.015, 0.015)$ meters. To implement the trained policy on a real-world robot, the output pth model generated by PPO was converted to the ONNX format and used by the ONNX-Runtime on the NUC.

B. The role of spinal rotation in trotting

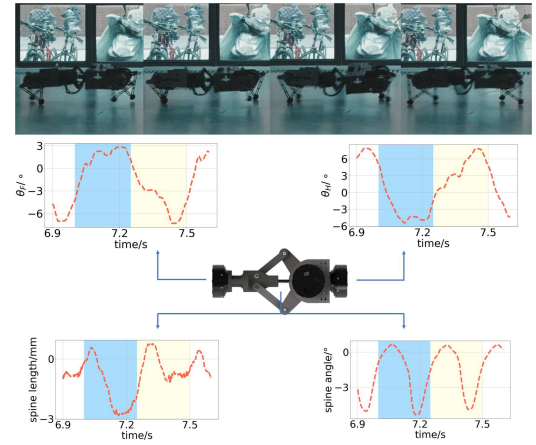
To verify the functionality of every joint on STRAY, a trotting gait with pure trajectory based control was first implemented. The gait involves simple tracking of the foot trajectory shown in the figure 4(a) using leg motors, while the desired positions and velocities of all spine motors are set to 0. As a result, the spine acts as a combination of passive linear and torsional springs and dampers between the front and the rear bases. However, direct deployment this way led to experimental failure because the feet could not leave the ground to enter the swing phase due to lateral body tilting caused by the stance leg on the other side of the base. Increasing the corresponding PD gains of the spine motors could not resolve the problem effectively.

To address this issue, the front and hind rolling motors of the spine are actively controlled by the learned policy, while the control weight ρ is set to 0.2. The spine angle and spine length are both set to the nominal values. The active twisting of the spine helps the front and back bases tilt laterally towards the stance leg, leaving enough space for the swing leg to lift its feet and extend forward. With this setup, STRAY successfully realizes a trotting gait with a desired velocity of 0.8 m/s, as shown in Figure 5. It can be seen that the two rolling motors within the spine actively help STRAY achieve a steady-state velocity of approximately 0.6 m/s.

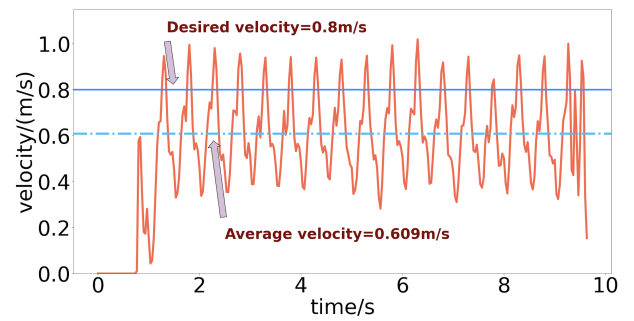
C. The role of spinal extension and rotation in bounding

In the actual implementation, because the bounding gait is symmetric laterally, the desired position for the front and rear spine rolling motors are set to 0. The spine length and angle are actively controlled by the learned policy, while the four feet follow the trajectory shown in Figure 4(b).

1) *Simulation*: After training, bounding gaits at five different speeds are obtained and tested in MuJoCo, ranging from 0.5 m/s to 2.0 m/s with an increment of 0.5 m/s. We



(a) Snapshots of STRAY trotting in the real world with the corresponding spine angle variations.



(b) Average forward velocity results from experiments.

Fig. 5: Snapshots of the 0.8 m/s trotting gait in experiments and the corresponding spine angle variation within a single period. The actual average forward velocity is approximately 0.6 m/s

also compare the effect of passive and rigid spines on the performance of the robot, as well as different values of the control weight ρ , the results of which can be found in Figure 6. Figure 6 presents the robot's performance in terms of average forward velocity and Cost of Transport (COT). It can be seen that when the robot uses active spinal action guided by the learned policy with the same control weight ρ , it can achieve the desired speed more effectively and efficiently, resulting in more accurate average forward velocities and lower COTs. Additionally, the hybrid controller with a higher control weight ρ of 0.2 performs better than that with ρ of 0.1, indicating the effectiveness of the proposed hybrid controller and the RL based portion inside.

2) *Experiments*: Preliminarily, the 1.0 m/s bounding gait with a control weight ρ of 0.2 is selected and implemented on STRAY after the sim-to-real transfer process, the results of which are shown in Figure 7. It can be found from Figure 7 that, during a complete period of the bounding gait, when the front legs lift off the ground, marked by the blue region, the spine extends to push the front base

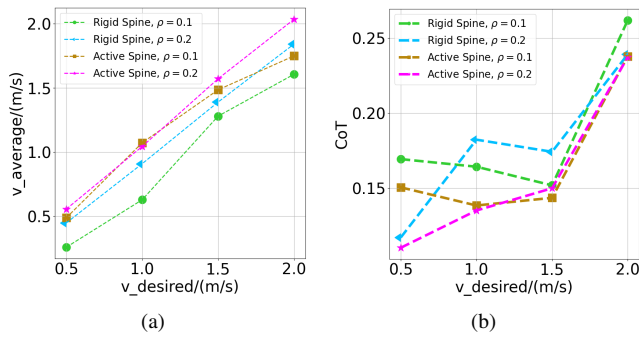
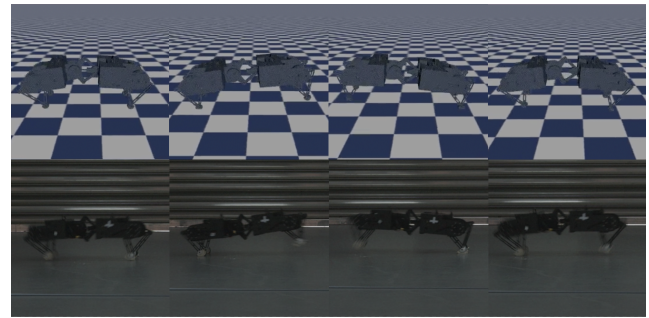


Fig. 6: Bounding performance of STRAY in simulation, average forward velocity and Cost of Transport achievable under different conditions. (a) Results on the average forward velocity. (b) Results on the Cost of Transport.

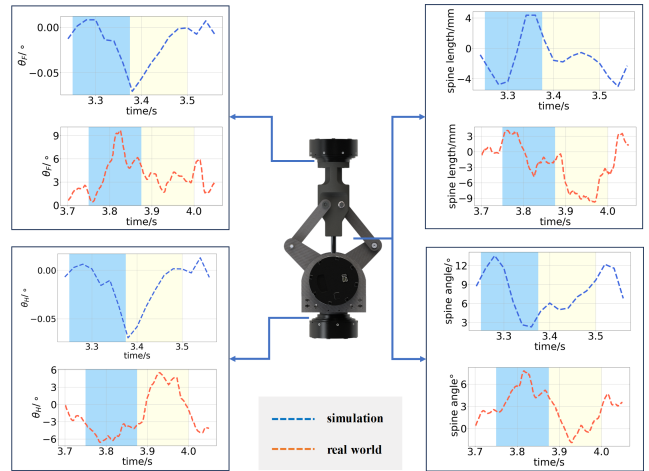
forward, and when the front legs touch the ground, marked by the yellow region, the spine retracts to pull the rear base forward. On the other hand, the spine angle first reduces and then increases during a single period, meaning the spine first bends upward and then downward to help the front base to actively move forward. The subtle movement of the two rolling motors is due to passivity and is negligible. It is worth noting that this pattern of spine movement seems simple and straightforward, but it matches our intuition and more importantly it is the natural result from the robot’s repetitive attempts to interact with the environment in the RL training process. This suggests that the spine extension and rotation plays a crucial role during the bounding gait, particularly at higher speeds to resonate with the body’s locomotion tempo to achieve stable and efficient performance. The selected policy for 1.0 m/s bounding gait successfully helps STRAY achieve a steady-state average forward velocity of 0.73 m/s. The learned policies for the other bounding velocities will go through the sim-to-real transfer process and be tested in the near future, but they will no doubt expose the robot to harsh experimental situations that could potentially lead to frustrating mechanical failures. Additional care is required through that process.

V. CONCLUSIONS

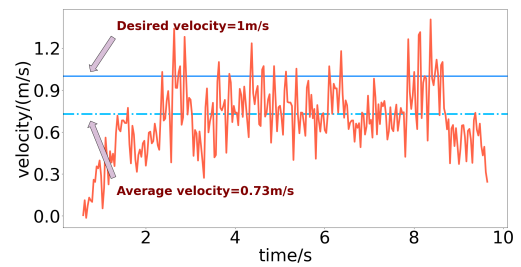
This paper presents the design of a four-DOF active spine and its incorporation into a quadrupedal robot STRAY. An RL based hybrid controller has been developed to actively control the spine’s motion for trotting and bounding gaits. The controller successfully helps STRAY realize dynamic locomotion in both simulation and experiments, with average forward velocities of 0.6 m/s and 0.7 m/s, respectively. It is found that actively twisting the spine during the trotting gait and extending and rotating the spine during the bounding gait can improve the dynamic performance of STRAY, with more accurate velocity tracking and higher locomotion efficiency. More interestingly, the RL based controller, as the natural results learned from robot-environment interaction, indicates that the spine’s extension and rotation are quite helpful for bounding gaits, especially at high speeds



(a) Snapshots of STRAY bounding in MuJoCo and the real world.



(b) Spine angle variations in simulation and experiments.



(c) Average forward velocity results from experiments.

Fig. 7: Snapshots of the 1.0 m/s bounding gait in simulation and experiments. The corresponding spine angle variations within a single period are also presented, as well as the actual average forward velocity.

For the future work, we plan to extend the experimental validation for the bounding gaits and further improve STRAY’s dynamic performance. Also, instead of treating the extension and rotation of the spine as a passive spring, these two motors will be utilized to achieve better trotting gaits. Eventually, all the DOFs within the spine will be utilized together to enable STRAY to realize agile locomotion behaviors and achieve advance mobility on diverse terrains.

REFERENCES

- [1] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and 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)*, 2018, pp. 2245–2252.
- [2] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, *et al.*, "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.
- [3] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6295–6301.
- [4] G. Kenneally, A. De, and D. E. Koditschek, "Design principles for a family of direct-drive legged robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 900–907, 2016.
- [5] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.
- [6] R. M. Alexander, *Principles of animal locomotion*. Princeton university press, 2003.
- [7] M. Hildebrand, "Motions of the running cheetah and horse," *Journal of Mammalogy*, vol. 40, no. 4, pp. 481–495, 1959.
- [8] J. Duperret and D. E. Koditschek, "Empirical validation of a spined sagittal-plane quadrupedal model," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1058–1064.
- [9] J. Duperret, B. Kramer, and D. E. Koditschek, "Core actuation promotes self-manipulability on a direct-drive quadrupedal robot," in *2016 International Symposium on Experimental Robotics*. Springer, 2017, pp. 147–159.
- [10] J. D. Caporale, Z. Feng, S. Rozen-Levy, A. M. Carter, and D. E. Koditschek, "Twisting spine or rigid torso: Exploring quadrupedal morphology via trajectory optimization," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1177–1184.
- [11] S. Bhattacharya, A. Singla, D. Dholakiya, S. Bhatnagar, B. Amrutur, A. Ghosal, S. 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)*. IEEE, 2019, pp. 1–6.
- [12] W. Li, Z. Zhou, and H. Cheng, "Dynamic locomotion of a quadruped robot with active spine via model predictive control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1185–1191.
- [13] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.
- [14] J. M. Brown, J. L. Pusey, and J. E. Clark, "Design methodology of linkage morphology for high speed locomotion," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2017, pp. 730–736.
- [15] E. Todorov, T. Erez, and 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.
- [16] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: <http://jmlr.org/papers/v22/20-1364.html>
- [17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.