2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

# Spine dynamics as a computational resource in spine-driven quadruped locomotion

Qian Zhao, Kohei Nakajima, Hidenobu Sumioka, Helmut Hauser, and Rolf Pfeifer

*Abstract*— **Recent results suggest that compliance and nonlinearity in physical bodies of soft robots may not be disadvantageous properties with respect to control, but rather of advantage. In the context of morphological computation one could see such complex structures as potential computational resources. In this study, we implement and exploit this view point in a spine-driven quadruped robot called Kitty by using its flexible spine as a computational resource. The spine is an actuated multi-joint structure consisting of a sequence of soft silicone blocks. Its complex dynamics are captured by a set of force sensors and used to construct a closed-loop to drive the motor commands. We use simple static, linear readout weights to combine the sensor values to generate multiple gait patterns (bounding, trotting, turning behavior). In addition, we demonstrate the robustness of the setup by applying strong external perturbations in form of additional loads. The system is able to fully recover to its nominal gait patterns (which are encoded in the linear readout weights) after the perturbation has vanished.**

## I. INTRODUCTION

Traditional robots use rigid materials for structural elements and for actuators, e.g., for their body, arms, and motors. Such rigid body and high torque servos are widely used to allow for precise control and to suppress unwanted dynamics. Although this approach has successfully demonstrated its applicability for achieving various tasks, it requires intensive computation as every degree of freedom has to be precisely controlled at every single time step. Furthermore, these robots perform much worse and less naturally compared to their biological counterparts. In contrast, robots with compliant bodies, could solve this problem, for instance, by applying biologically inspired design to robots to facilitate, e.g., locomotion, while using simple controller [1]–[5]. This indicates that part of the computation need for control can be outsourced to the body by using suitable morphological properties. In this sense, a compliant body may not be a factor to make control hard. Instead, it could be a potential computational resource.

This hypothesis, usually referred to as morphological computation (MC), has recently received some theoretical support by Hauser et al. [6], [7]. They proposed theoretical models for MC with compliant bodies, where they demonstrate how compliant physical bodies can be potentially used as a computational resource. They applied the concept of reservoir computing[1] to random networks of mass-spring systems. Instead of using a neural network or a network of leaky integrators (as in standard approaches for reservoir computing) the previously mentioned models employ a compliant physical body as a reservoir. The theory suggests that complex physical bodies of soft robots could be a potential computational resource, due to their elasticity and nonlinearity inherently embedded in their physical bodies. Additionally, Hauser et al. [7] demonstrated how static feedback from the sensor into the physical body (via actuators) can be used to generate autonomously periodic patterns [5], e.g., as used in locomotion.

In this context, there have been some successful examples of the implementation of the concept of MC. In one case, a simple model of a human musculoskeletal system was used to identify the capacity of computation [9]. In a more biologically plausible example, the computational capacity of a muscular-hydrostat system was investigated and found to have a characteristic memory capacity [10]. In addition, such a system has been demonstrated to have the potential to emulate complex nonlinear dynamical systems, and closed-loop controls [11]. In terms of locomotion, a simulated tensegrity robot has been demonstrated to be capable of embedding nonlinear limit cycles based on different online learning techniques [12]. However, these works are all limited to simulators based on predefined environments and precise and sufficient data collection.

In this study, we implemented this theoretical model to a real spine-driven quadruped robot called Kitty. The impact of real-world conditions on the physical reservoir will be considered, including the partial loss of the state of the morphology, noisy sensory time series, and limited training phase. The spine embedded in Kitty robot is an actuated multi-joint structure consisting of a sequence of compliant silicone blocks and its dynamics is captured by a set of force sensors [13]. Its design is inspired by the biological hypothesis of spinal engine stating that locomotion is mainly achieved by the spine, while the legs may serve as assistance [14], [15].

In this paper, we first introduce a biologically-inspired

This research was funded in party by the European Community's Seventh Framework Programme FP7-ICT-248311 (AMARSi) and FP7-ICT-231688 (Locomorph).

Q. Zhao, K. Nakajima, H. Hauser, and R. Pfeifer are with Department of Informatics, University of Zurich, 8050, Zurich, Switzerland, email: {zhao/nakajima}@ifi.uzh.ch

K. Nakajima is also with Bio-Inspired Robotics Lab, Swiss Federal Institute of Technology Zurich, 8092 Zurich, Switzerland.

H. Sumioka is with the Advanced Telecommunications Research Institute International.

[1]Reservoir computing is a machine learning technique used to emulate complex, nonlinear computations by employing a randomly initiated (but afterwards fixed in their parameters) complex, nonlinear dynamical network of nonlinear dynamical systems (i.e., the reservoir). For more details we refer to [8].

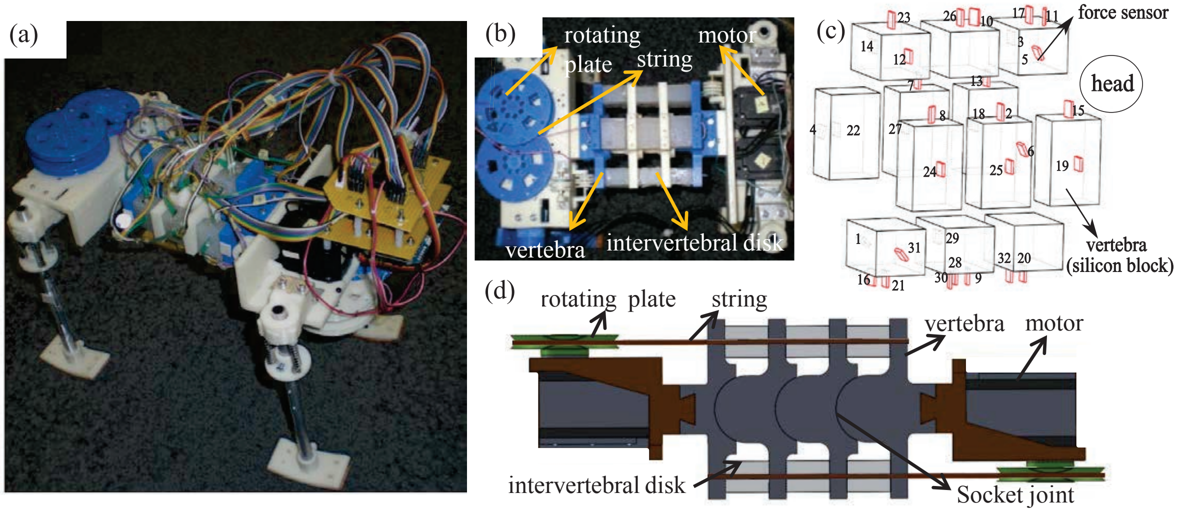


Fig. 1. (a) A quadruped robot equipped with a tendon-driven spine. (b) A biologically inspired spine. (c) The arrangement of 32 force sensors in the spine. Cubic with red contour stands for the force sensor, while cubic with black contour indicates silicone block. (d) Cross section of the artificial spine: sagittal view.

multiple degree-of-freedom spine model [16], [17] to a real quadruped robot and explain its potential to be a computational resource. Second, the experimental procedures are described, including the overall information processing based on MC. Then, experimental results by using actual sensory data from physical robotic system are presented. The results suggest that with the help of the complaint spine (as a computational resource) this MC setup is able to encode movement patterns, produce rhythmic patterns, and learn new pattern. Finally, the robustness of this learned behavior against external perturbation is investigated. We found that noises coming from the real-world conditions benefit the robustness of such a system.

## II. Robot Design

Kitty is equipped with a flexible spine (29 cm wide, 32 cm long, 20 cm high, and 1.4 kg) (Fig. 1 (a)). Three springs are mounted in each stick-shaped leg to cushion the shock from the ground. The legs are fixed to the body and have no relative rotation with respect to it. The bottoms of the feet are glued with asymmetrical friction material to guide the walking direction.

Figure 1 (b) shows an artificial spine endowed with biological characteristics. It consists of cross-shaped rigid vertebrae made of ABS plastic, intervertebral disks made of silicone blocks and strings driven by motors, similar to the anatomical spine structure [18]. As shown in Fig. 1 (b), (d), the vertebrae are separated by the silicone blocks, which work as intervertebral discs, and connected by four strings through themselves and the silicone blocks. The four strings connecting vertebrae and intervertebral disks are pulled respectively by four RC motors, which can control the movements of the spine.

Motor command $I_i(t)$ to motor $i$ for the spine movements is computed with sinusoid function given by:

$$I_i(t) = A\sin(2\pi f_i t + \phi_i) + \psi \quad i = \{u,d,r,l\}, \tag{1}$$

where $i$ stands for the position of the motor mounted in the robot. $u$, $d$, $r$, and $l$ indicate motors controlling the strings located at up, down, right, and left side of the spine, respectively.

The dynamics of the spine is captured by 32 force sensors (FSR400) randomly embedded into silicone blocks (Fig. 1 (c)). The dynamics of this spine is complex due to its flexibility and compliance. In addition, according to the geometrical constraint of the spine configuration, it exhibits highly nonlinear dynamics during locomotion. This suggests the possibility of this compliant spine to be a computational resource and generate diverse locomotive behaviors.

## III. Information processing in flexible spine

The task of Kitty is to generate locomotive behaviors by using the dynamics of its spine. The pre-designed motor commands are emulated by static, linear readout units after learning. Figure 2 shows an overview of the information processing based on MC. The robot dynamics are generated by the spine movements driven by four motors. One pattern generator corresponds to one specific locomotive behavior. It consists of four linear readout units (blocks in the area highlighted in grey in Fig. 2) which are associated with motors controlling the up, down, right, left side of the spine. The states of the spine are measured through randomly distributed force sensors (FSR400) in the silicone blocks. Because it is unclear which arrangement of the sensors is the best to perform tasks, we adopted random topology similar to original echo state network.

MC consists of three phases: teaching, learning, and evaluating phases. We take one specific gait G, where $G \in$

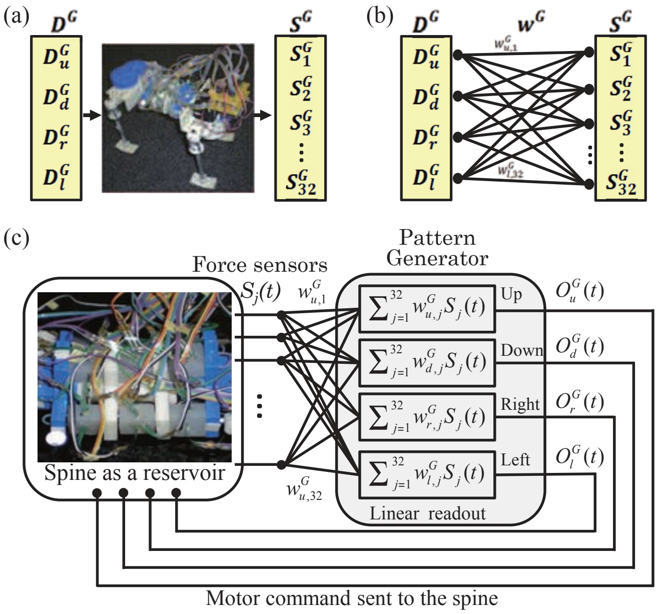


Fig. 2. An overview of information processing in the system. (a) Teaching phase (open loop): the predefined motors command $\mathbf{D^G}$ are sent to drive the robot and the associated sensory responses $\mathbf{S^G}$ are collected; (b) learning phase (open loop): linear readouts $\mathbf{w_i^G}$ are adjusted to emulate desired outputs; (c) evaluating phase (close loop): the motor commands $O_i^G(t)$, where $i \in \{u,d,r,l\}$ are generated by a physical spine and sent to the motors to drive the spine. The motor commands are computed as a sum of current states of force sensors multiplied with output weights $\mathbf{w_i^G}$.

$\{bounding, trotting, turningLeft\}$, as an example to explain the information process.

The teaching phase is implemented in open loop where the motor commands (Eq.1) are sent to the robot and drive it (Fig. 2 (a)). The target signals for four motors in one gait are stored as a vector $\mathbf{D^G} = (\mathbf{D_u^G}, \mathbf{D_d^G}, \mathbf{D_r^G}, \mathbf{D_l^G})$, where $u$, $d$, $r$, $l$ indicate upside, downside, right side, and left side of the spine, respectively. Accordingly, the associated state $S_j(t)$ of force sensor $j$ at every time step $(t = 1, 2, \cdots, M)$ is collected in a $M \times N$ matrix, $\mathbf{S^G}$, where $N = 32$ (the number of force sensors), and $M$ is the time step.

The learning process is carried out with open loop (Fig. 2 (b)). In the learning phase, only the linear readouts are adapted, i.e., $\mathbf{w_i^G}$ are adjusted. The system is forced into the desired motor commands by a "teacher" signal. Therefore, the optimal output weights $\mathbf{w_i^G}$ are calculated by $\mathbf{w_i^G} = (\mathbf{S^G})^+\mathbf{D^G}$, where $(\mathbf{S^G})^+$ stands for the (Moore-Penrose) pseudo-inverse of $\mathbf{S^G}$.

In the evaluating phase, the loops are closed (Fig. 2 (c)). Spine dynamics are sent to linear and static readout units that compute outputs of the system $O_i^G(t)$, where $i \in \{u,d,r,l\}$. For each linear readout, $O_i^G(t)$ is a sum of values of the force sensors $S_j(t)$ multiplied by output weights $\mathbf{w_i^G} = (w_{i,1}^G, w_{i,2}^G, \cdots, w_{i,32}^G)$: $O_i^G(t) = \sum_{j=1} w_{i,j}^G S_j(t)$. In this formula, $w_{i,j}^G$ indicate the output weight for j-th force sensor for linear output i, and $S_j(t)$ is the value of the j-th sensor at time $t$.

## IV. Experimental Setting

### A. Teaching signals

The target signals in this physical reservoir computing are the commands sent to the motors located at the up, down, right, and left side of the spine, which control spine movements. The spine of Kitty robot is controlled with periodic motor commands given by Eq.1 using parameters shown in Table I. Note that motors whose parameters are marked with asterisks in the table are controlled with constant values to keep the natural length without stretching and relaxing.

TABLE I

CONTROLLERS FOR SPINE MOVEMENTS

| controller | $(f_u, \phi_u)$ | $(f_d, \phi_d)$ | $(f_r, \phi_r)$ | $(f_l, \phi_l)$ |
|---|---|---|---|---|
| $S_{bounding}$ | $(\frac{1}{\pi}, 0.0)$ | $(\frac{1}{\pi}, \pi)$ | * | * |
| $S_{trotting}$ | * | * | $(\frac{1}{\pi}, \pi)$ | $(\frac{1}{\pi}, 0.0)$ |
| $S_{turningLeft}$ | $(\frac{1}{\pi}, 0.0)$ | $(\frac{1}{\pi}, \pi)$ | $(\frac{1}{\pi}, 0.0)$ | $(\frac{1}{\pi}, \pi)$ |

The bounding gait, as a result of spine flexion-extension movements, is generated by the controller $S_{bounding}$ [13]. This controller only pulls the strings located at the upside and downside of the spine alternately, while the side strings are kept the natural length. Similarly, the trotting gait, generated by spine lateral movements, can be achieved by controller $S_{trotting}$. It drives the motors on the left side and right side alternately. Turning left behavior, controlled by $S_{turningLeft}$, can be realized by combining bounding gait and trotting gait together. The only difference between turning right and left is the flip between $\phi_r$ and $\phi_l$, which are the phase lags with respect to the upside of the spine.

### B. Experimental procedures

To achieve MC, three phases are used: teaching, learning, and evaluating phases. In the teaching phase, the number of teaching data used to train the reservoir readouts is 600 time steps after initial 200 samples are discarded as transients. The number of sampling data for one cycle is heuristically set to 20.

## V. Results

In this section, the behavior of the physical reservoir is observed first. Then three different resultant locomotive patterns are analyzed. In the end, robustness of this reservoir is studied by adding external load on the robot.

### A. Versatile behaviors using the same physical body

Figure 3, 4 show the sequential pictures of bounding gait, turning left behavior in the evaluating phase, as a result of pronounced spine movements. The signals generated by such a spine reservoir are able to drive the robot to emulate a specific gait.

Figure 5, 6, and 7 show the best performance of the spine reservoir, associated sensory response, and footfall pattern in bounding gait, trotting gait, and turning left behavior, respectively. We found that the generated control signals are periodic and similar to the desired ones in terms of the frequency and the shape (Fig. 5 (a), 6 (a), and 7 (a)).

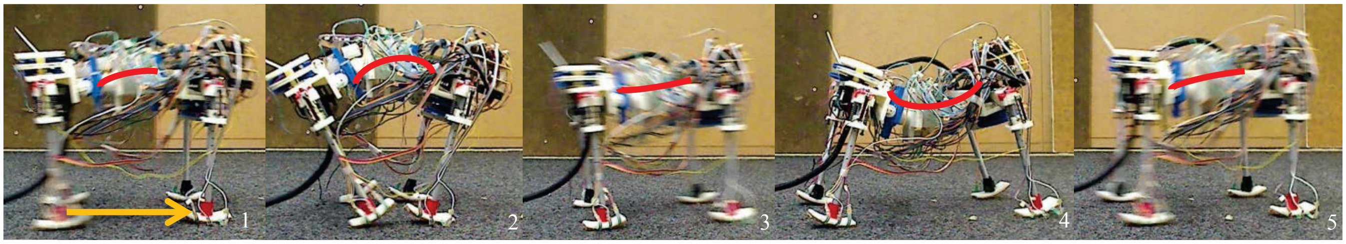


Fig. 3. Sequential pictures of spine movements in the bounding gait in the evaluating phase. Orange arrow represents the walking direction.

Fig. 4. Sequential pictures of spine movements in turning left behaviour in the evaluating phase. Orange arrow represents the walking direction.

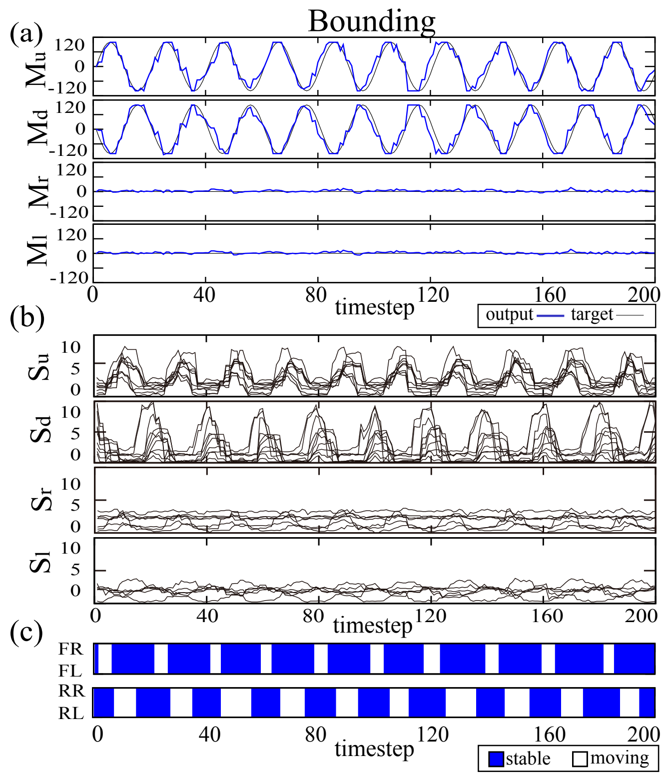


Fig. 5. (a) The performance of pattern generator in evaluating phase: bounding gait. Four subplots from top to bottom are the results of pattern generators for motor controlling the up, down, right, left side of the spine, respectively. Y-axis stands for the amplitude sent to the motor. X-axis indicates the time steps. The grey thin curve is the target trajectory and the blue thick curve is the actual output from the spine reservoir. (b) Sensory responses in evaluating phase. Four subplots from top to bottom are sensory responses collected at the up, down, right, left side of the spine, respectively. Y-axis stands for the force [N] measured from the sensors. (c) Footfall patterns in evaluating phase: bounding gait (FR: front right leg; FL: front left leg; RR: rear right leg; RL: rear left leg).

Figure 5 (b), 6 (b), and 7 (b) suggest that the dynamics of this spine has a specific correspondence to each behavior, i.e., dynamic of each side of the spine (sensory values of all sensors located in each side) has different patten according to each behavior. For instance, the sensors located at the upside and downside response more than the ones at the right and left side in the bounding gait, as a result of sagittal spine movements (Fig. 5 (b)). Lateral spine movements result in trotting gait. As a consequence, sensory responses at right and left side of the spine are much higher than the responses captured in the up and down side (Fig. 6 (b)). Since turning left behavior emerges when bounding and trotting gait are combined together, the sensory responses are also a combination from bounding and trotting gait (Fig. 7 (b)).

We also noticed that the actual signals cannot very precisely emulate the desired signals. This is due to the limitation of physical platform and arena. For example, the motors easily get hot and stop working after 2,000 teaching time steps, or the robot is sensitive to the terrain because of the lack of ground clearance. However footfall patterns clearly show that the legs are coupled correctly to achieve bounding, trotting, and turning left behaviors, even if little phase delay and error exist (Fig. 5 (c), 6 (c), and 7 (c)).[2] In this paper, we define the footfall pattern based on whether the feet move forward or keep stable, because Kitty robot does not have any actuation on the legs, especially knee joints which mostly contribute to lift up the feet and produce ground clearance.

Figure 8 shows the obtained readout weights for each gait. The weights are adjusted in the learning phase and then are fixed in the evaluating phase. We observed that in bounding gait (Fig. 8 (a)), the weights associated with the up-down motors have higher values than the ones with right-

[2]In this paper, we did not adopt measures evaluating the difference between the target and actual commands, such as Mean Square Error (MSE). This is because we often observed a case that the actual motor command generate a seemingly correct motor command with phase shift, compared with the target motor command. When we use MSE for example, this effect avoids the appropriate evaluation of the actual motor commands.

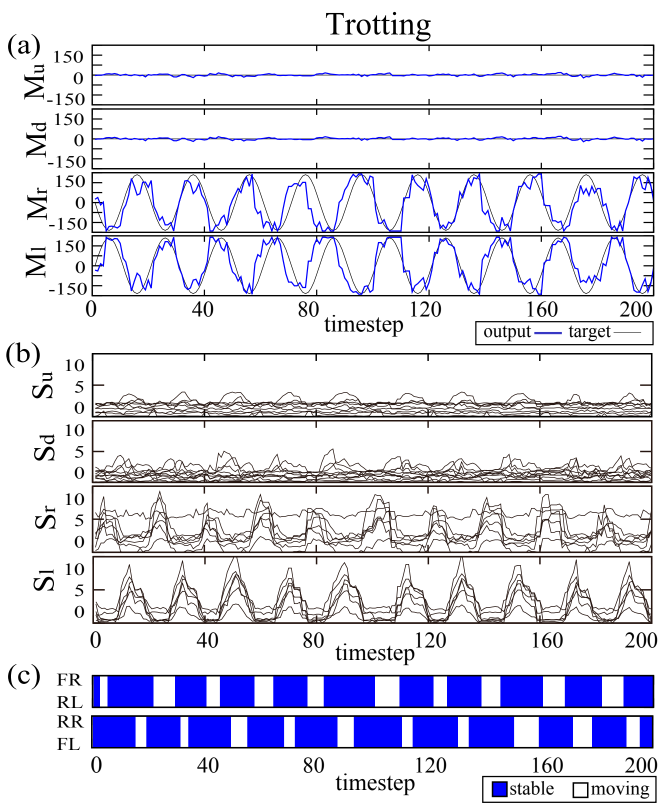


Fig. 6. (a) The performance of pattern generator in evaluating phase: trotting gait. X and Y axes in four subplots represent the same meaning as Fig. 5 (b) Sensory responses in evaluating phase. Y-axis stands for the force [N] generated by the sensors.(c) Footfall patterns in evaluating phase: trotting gait (FR: front right leg; FL: front left leg; RR: rear right leg; RL: rear left leg).

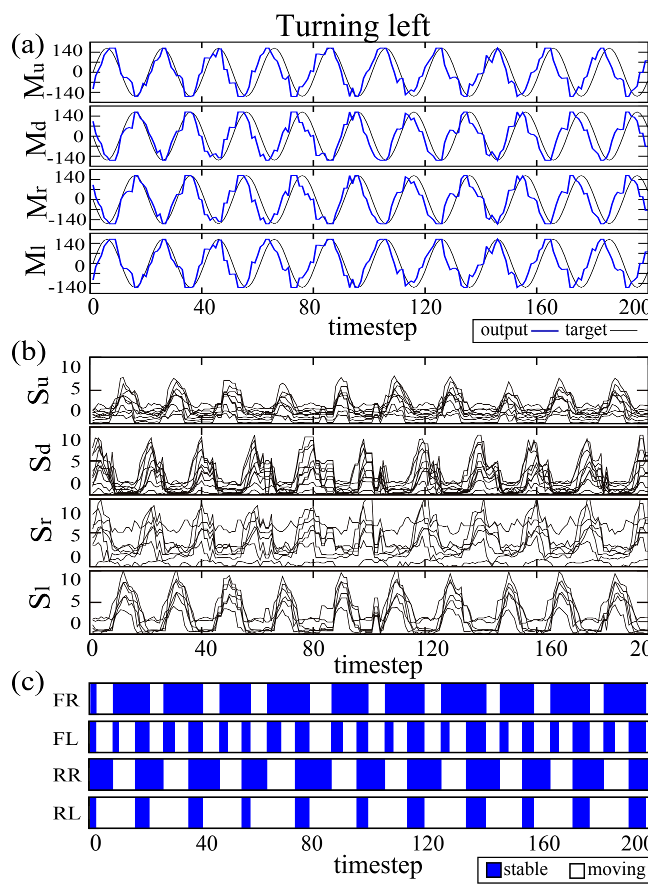


Fig. 7. The performance of pattern generator in evaluating phase: turning left behavior. X and Y axes in four subplots represent the same meaning as Fig. 5 (b) Sensory responses in evaluating phase. Y-axis stands for the force [N] generated by the sensors. (c) Footfall patterns in evaluating phase: turning left behavior (FR: front right leg; FL: front left leg; RR: rear right leg; RL: rear left leg).

left motors. The motors controlling strings located at right and left sides do not contribute to the bounding gait so much. This is also reflected in the readout weights for right and left motors, whose values are close to zero and overlap with each other. The weights for up motors are nearly mirror image of the weights for down motors about dashed line. The results reveal that the weights can reflect the coordination among the motors. Similarly, in trotting gait, the weights of linear readouts for up and down motors are nearly zero, while the weights of linear readout for right and left motors are almost symmetrical about dashed line. In turning left behavior, the weights of linear readouts for right and down motors are overlapped, and the same for the rest two sets.

These results suggest that this compliant spine can be regarded as a computational device to generate repetitive movements, in addition to be a mechanical component connecting the front legs and rear legs. Indeed, multiple behaviors can be produced by the same physical body, only by adjusting the linear readouts.

### *B. Robustness against external perturbation in bounding gait*

One crucial criterion in evaluating the performance of learning is robustness against external perturbation. There-

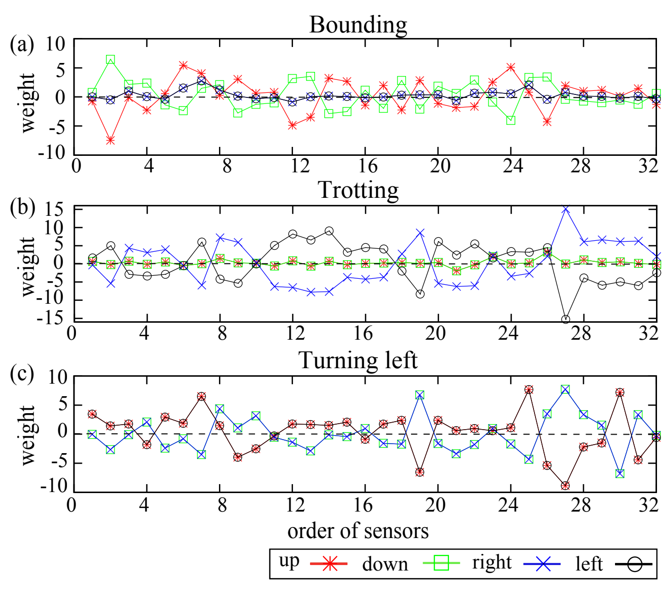


Fig. 8. The weights of linear readout trained in the learning phase in bounding gait (a), trotting gait (b), and turning left behavior (c). X axis stands for the order of the sensors, and y axis represents the value of weights.

fore, it is important to test how robust the spine reservoir is against external perturbation. We tested this characteristic by adding load in the front part of the body (Fig. 9 (a)). The number of evaluating phase is 360 time steps. In the first 40 time steps, the robot is moving without any load. At time step 40, an external load is added to the front body and remains until time step 200. Different loads ranging from 100 [g] to 1000 [g] have been tested in this spine reservoir, as shown in Fig. 9 (b). The experiments were conducted five trials. The average speed and the standard derivation were recorded when the external load is applied on the robot for 160 time steps. The results suggest that the speed of Kitty has a negative correlation with the external load. We observe that with the increase of the load, the performance of the spine reservoir gets affected more. In other words, the generated signals more easily get stuck at some points. Accordingly, the robot vibrates at these postures. This vibrating movements do not contribute to the speed too much and might account for the resultant slower speed. In addition, the stability gets worse with the increase of external loads.

Figure 9 (c) shows two typical cases: one is with load 400 [g] and the other is with 1000 [g]. In the former case, when the external load is added, the amplitude of the signals generated by the spine reservoir gets suppressed in the following three cycles, but the frequency still remains the same. From time step 100, the spine reservoir recovers its repetitive performance. After the load is moved, the performance of this physical reservoir get affected again and the amplitude drops. It starts to recover after two cycles. However in the latter case, the load is too heavy, almost two-thirds of Kitty robot's own weight. Thus, this load stops reservoir's performance. But once it is removed, the ability of emulating desired signal is restored and locomotive pattern continues. This good performance of the learned behavior against external perturbation might be accounted for the noise, which is inherent to Kitty robot.

Because robustness can be enhanced by manually added noise in teaching phase in the simulator [7], we think that the observed good performance against external perturbation might be accounted for noise, which is inherent to Kitty robot.

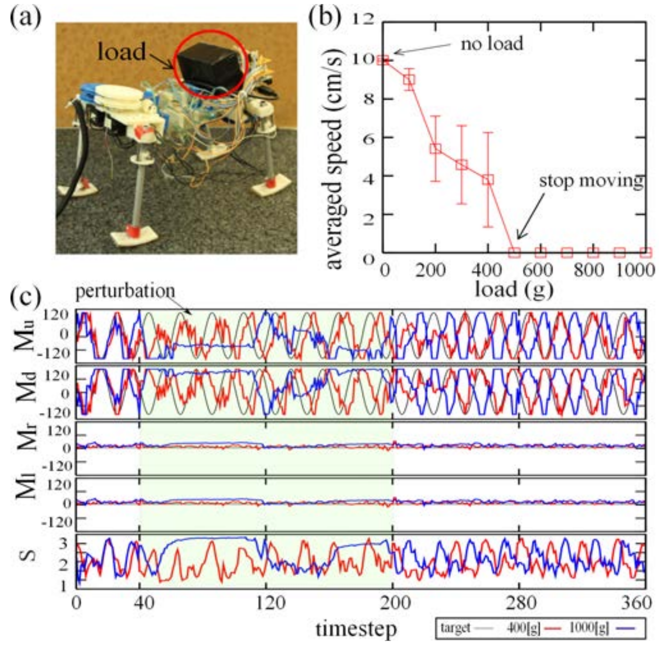


Fig. 9. (a) Kitty robot with external load. (b) Stability of spine reservoir against various external loads in bounding gait. (c) The performance of spine reservoir in evaluating phase with external load in bounding gait. The top four subplots are the comparison between target signal and actual output signal from reservoir for the motor controlling the up, down, right, left side of the spine, respectively. The bottom subplot is the dynamics of the spine which is average value over 32 force sensors. Red curve represents the case when external load weighing 400 [g] is applied, while blue curve is the case with the load of 1000 [g]. Grey curve is the target signal. X-axis indicates the time steps. The area marked in green is the period when the spine reservoir is disturbed. In the top four subplots, Y-axis stands for the amplitude [degree] sent to the motor. In the bottom subplot, Y-axis stands for the force [N].

## VI. Discussion and Conclusion

In this paper, we demonstrated that the developed compliant spine structure is not only a mechanical component connecting the front legs and rear legs, but rather can also serve as a computational resource to achieve different behaviors, such as bounding gait, trotting gait, and turning behavior. The results suggest that this computational resource (compliant spine) together with linear and static readouts and feedback loops is able to encode movement patterns, produce rhythmic patterns, and learn new pattern. Remarkably, multiple behaviors can be produced by the same fixed physical body, simply by readjusting the weights of the linear readouts. In addition, we demonstrated the robustness of the learned behavior by applying additional load as external perturbations. The results show that this system is able to recover to its nominal gait patterns encoded in the linear weights after the perturbation has disappeared.

In our experiments, although we were able to observe a successful locomotion for each gait pattern, the produced motor commands were noisy and were unable to emulate the target commands precisely. This is mainly caused by the limitation of this platform when it starts to interact with the environment. For instance, the amount and quality of the teaching data available in physical platform is much lower than in comparable simulated work. In spite of it, the spine reservoir was still able to produce stable and seemingly periodic locomotive patterns.

A possible solution to reduce the error between the target and the output signals is to optimize the spine structure by employing more asymmetrical features. This is inspired by biology, where asymmetrical spine structure can be observed in animals. Such features are unevenly distributed muscles in the spine, asymmetrical muscle stiffness, and the shape of the spinal column [19]. These properties could increase the diversity and nonlinearity of the spine reservoir and improve its performance. In addition, in order to better reflect the spine dynamics in the sensory time series, the number and the locations of the sensors within the spine can be explored and optimized in future work.

Another future research direction is to ask whether a gait switching could be achieved in our framework [20], [21]. One possible scenario would be to explore ways to embed multiple gaits with a single fixed linear readout and a feedback loop by introducing an input signal corresponding to each gait to the spine. The signal acts as an initiation signal (or a control signal) for the gait switching and would be provided either as an external or internal control command [7]. Especially in our contexts, the signals can be mechanical, such as an intentional movements of a head or a tail of the robot's body, or can be also generated from an environmental change, such as the change of the terrain.

## REFERENCES


[1] F. Iida and R. Pfeifer, "Sensing through body dynamics," *Robotics and Autonomous Systems*, vol. 54, no. 8, pp. 631–640, 2006.

[2] S. Kim, J. E. Clark, and M. R. Cutkosky, "isprawl: Design and tuning for high-speed autonomous open-loop running," *The International Journal of Robotics Research*, vol. 25, no. 9, pp. 903–912, 2006.

[3] K. Hosoda, T. Takuma, A. Nakamoto, and S. Hayashi, "Biped robot design powered by antagonistic pneumatic actuators for multi-modal locomotion," *Robotics and Autonomous Systems*, vol. 56, no. 1, pp. 46–53, 2008.

[4] R. Pfeifer and J. Bongard, *How the Body Shapes the Way We Think: A New View of Intelligence*. The MIT Press, 2006.

[5] R. Pfeifer, M. Lungarella, and F. Iida, "Self-organization, embodiment, and biologically inspired robotics," *Science*, vol. 318, no. 5853, pp. 1088–1093, 2007.

[6] H. Hauser, A. Ijspeert, R. Fuechslin, R. Pfeifer, and W. Maass, "Towards a theoretical foundation for morphological computation with compliant bodies," *Biological Cybernetics*, vol. 105, pp. 355–370, 2011.

[7] ——, "The role of feedback in morphological computation with compliant bodies," *Biological Cybernetics*, vol. 106, pp. 595–613, 2012.

[8] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: theory, applications and implementations," in *Proceedings of the 15th European Symposium on Artificial Neural Networks*, 2007, pp. 471–482.

[9] H. Sumioka, H. Hauser, and R. Pfeifer, "Computation with mechanically coupled springs for compliant robots," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, Sept., pp. 4168–4173.

[10] K. Nakajima, H. Hauser, R. Kang, E. Guglielmino, D. G. Caldwell, and R. Pfeifer, "Computing with a muscular-hydrostat system," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 1496–1503.

[11] ——, "A soft body as a reservoir: case studies in a dynamic model of octopus-inspired soft robotic arm," *Frontiers in Computational Neuroscience*, vol. 7, no. 91, pp. 1–19, 2013.

[12] K. Caluwaerts, M. D'Haene, D. Verstraeten, and B. Schrauwen, "Locomotion without a brain: Physical reservoir computing in tensegrity structures," *Artificial Life*, vol. 19, no. 1, pp. 35–66, 2013.

[13] Q. Zhao, K. Nakajima, H. Sumioka, X. Yu, and R. Pfeifer, "Embodiment enables the spinal engine on quadruped robot locomotion," in *2012 IEEE/RSJ Internattional Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 2449–2456.

[14] S. Gracovetsky and S. Iacono, "Energy transfers in the spinal engine," *Journal of Biomedical Engineering*, vol. 9, no. 2, pp. 99–114, 1987.

[15] S. Gracovetsky, *The Spinal Engine*. Springer, cop, 1989.

[16] Q. Zhao, H. Sumioka, and 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 (AMAM2011)*, Oct. 2011, pp. 51–52.

[17] Q. Zhao and H. Sumioka, "The effect of robot morphology on locomotion from the perspective of spinal engine in a quadruped robot," in *International Conference on Morphological Computation*, Sep. 2011, pp. 130–132.

[18] R. Pashman and T. Kim, "Normal spinal anatomy," 2012. [Online]. Available: http://www.espine.com/anatomy-normal.htm

[19] R. M. Alexander and A. S. Jayes, "Estimates of the bending moments exerted by the lumbar and abdominal muscles of some mammals," *Journal of Zoology*, vol. 194, no. 3, pp. 291–304, 1981.

[20] D. Owaki, L. Morikawa, and A. Ishiguro, "Listen to body's message: Quadruped robot that fully exploits physical interaction between legs," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 1950–1955.

[21] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," *Science*, vol. 315, no. 5817, pp. 1416–1420, 2007.