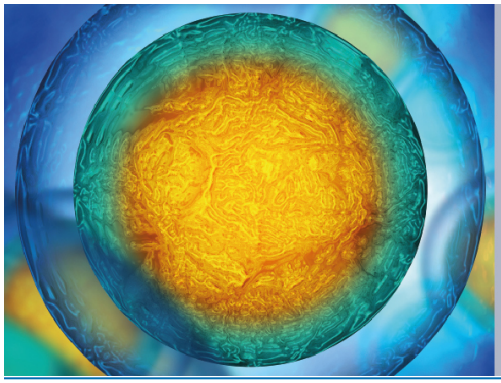
PAPER

Neurodynamic modeling of the fruit fly *Drosophila melanogaster*



Bioinspiration & Biomimetics

PAPER

Neurodynamic modeling of the fruit fly *Drosophila melanogaster*

RECEIVED
16 March 2020

REVISED
2 June 2020

ACCEPTED FOR PUBLICATION
18 June 2020

PUBLISHED
11 September 2020

C A Goldsmith, N S Szczecinski and R D Quinn
Case Western Reserve University, 10900 Euclid Ave, Cleveland, OH 44106, United States of America
E-mail: cag111@case.edu

Keywords: computational neuromechanics, synthetic nervous system, hexapod, legged robot, morphological modeling, Drosophila melanogaster

Abstract

This manuscript describes neuromechanical modeling of the fruit fly *Drosophila melanogaster* in the form of a hexapod robot, Drosophibot, and an accompanying dynamic simulation. Drosophibot is a testbed for real-time dynamical neural controllers modeled after the anatomy and function of the insect nervous system. As such, Drosophibot has been designed to capture features of the animal's biomechanics in order to better test the neural controllers. These features include: dynamically scaling the robot to match the fruit fly by designing its joint elasticity and movement speed; a biomimetic actuator control scheme that converts neural activity into motion in the same way as observed in insects; biomimetic sensing, including proprioception from all leg joints and strain sensing from all leg segments; and passively compliant tarsi that mimic the animal's passive compliance to the walking substrate. We incorporated these features into a dynamical simulation of Drosophibot, and demonstrate that its actuators and sensors perform in an animal-like way. We used this simulation to test a neural walking controller based on anatomical and behavioral data from insects. Finally, we describe Drosophibot's hardware and show that the animal-like features of the simulation transfer to the physical robot.

1. Introduction

Biology is a natural place to look for inspiration when building legged robots. Animals provide examples of how machines could be built to traverse difficult terrain with speed and agility. Despite how much is known about animal neuromechanics, there are still many open questions regarding how animals move. Computational neuromechanics [80] and biomimetic robots [35, 51] provide opportunities to organize what is known about animal locomotion, consolidate many different results into one model, observe how complete the current body of knowledge is, test hypotheses, and propose future experimental work. In this manuscript, we conceptualize, simulate, and build a hexapod robot as a computational neuromechanical model of the fruit fly. We use a simulation to verify that our modeling methods capture key animal-like responses and behaviors, and then we show that the robot can capture these same features.

Many robots developed over the past few decades exploit varying degrees of biological inspiration to control their locomotion [11, 36]. Early systems like Raibert's Quadruped and CWRU's Robot II utilized simplified biological gaits and joint constraints to successfully emulate biological motion [22, 48]. Contemporary robots have built upon the concept of biological gaits with additional success [5, 7, 53]. Some robotic platforms have expanded upon gait generation further and use biologically observed mechanisms such as central pattern generators (CPGs) in their controllers [19, 21, 23, 37, 41, 56, 63, 74].

Additional robots also mimic the mechanics of walking animals. One way to mimic animal mechanics is to incorporate compliant elements into the limbs and actuators. Compliant structures mimic the natural elasticity of muscle and tendon, enabling the robot to passively adapt to external forces or imprecise foot placement [26, 28, 34, 47, 60, 74]. Compliant structures of a known stiffness in series with actuators may enable the robot to measure output forces as in a series elastic actuator (SEA) [20, 52]. Elastic elements in parallel with the actuators can reduce the load on actuators by offloading torque as the element is deflected, aiding in control and potentially extending the actuators' lifespan [42, 74]. Some actuators are themselves highly compliant, such as braided pneu-

matic actuators, which must be used in antagonistic pairs like muscle–tendon complexes [2, 40, 54, 59]. Despite their added mechanical complexity, elastic components may increase the biomimicry, capability, and durability of legged robots.

Many of these robots represent a functional approach to biomimicry; their designs are focused on replicating end behaviors of the animal rather than beginning with the basic organization of the systems that produce them [11]. This strategy successfully improves robotic locomotion; however, it abstracts the morphology of the animal so that the robotic and biological structures are only trivially comparable. As our current understanding of animal locomotion is incomplete, this approach misses out on the potential for greater discovery; biologically inspired robots with higher levels of animal fidelity serve as testing platforms for biological hypotheses, advancing both fields. This is called the morphological approach [11].

An inexpensive and rapid starting point for the morphological approach is to simulate the mechanics of a robot and its controller [17, 20, 74]. The simulation provides an idealized environment for conceptual testing, which can be used to refine the robot's design for construction. Building and testing on the robot then provides further insight into these concepts by introducing internal (friction, noise, timing delay) and external (uneven terrain, ground slippage) variability that more closely resembles what the animal contends with, but are complicated and computationally expensive to accurately model in simulation. Consequently, some features prove significantly easier to model with a robot than in a simulation. For example, modeling limb strain in simulation necessitates adding additional degrees of freedom (DOF) and performing computationally expensive finite element analysis, while such data can be directly measured on the physical robot. As such, developing a physical platform is a crucial step in the morphological modeling of animals.

Over the last decade, the morphological approach has been applied to an increasing number of robots [20, 26, 42, 63, 74]. Many such robots, including HECTOR, Octavio, MantisBot, AMOS II, and BILL-Stick emulate insects. Insects are good candidates for morphological robots due to their robust locomotion capability with nervous systems that are smaller than those of other animals, such as mammals. This relative smallness might allow neurobiologists to understand the full insect nervous system before the much larger systems in vertebrates. Additionally, many of the mechanisms observed in the nervous systems of various historically studied insects such as cockroaches, locusts, and stick insects are both consistent across insects and can also be observed in phylogenetically distant animals [3, 50]. These findings imply that insect related discoveries could be applied to answer broader questions regarding animal neuromechanics.

Lately, fruit flies have become a much-studied model organism due to the abundance of genetic tools available to manipulate their nervous systems; for example, breeding strains that fail to express particular neuromodulators [81] or optogenetically suppressing sensory feedback during walking [45]. These and similar experiments have great potential to explain the neural control of walking in fruit flies. What has been discovered also appears to have strong parallels to what is known in other insects; for example, chordotonal organ (CO) afferents appear to encode the same quantities in stick insects and fruit flies [43, 57]. While much of the fruit fly nervous system is currently still unexplored, we believe the insects have unique potential to improve understanding of insect locomotion and broader locomotion control, making them valuable to additionally consider in robotic design.

Our goal is to apply the morphological approach to robotic design in the creation of an insect robot, modeling the mechanical and control features shared by a variety of insects to better leverage experimental findings for biological and robotic discovery. In particular, we aim to model the animal's morphology in four ways:

(a) Dynamic, morphological control to test detailed hypotheses about neural control in animals and exploit neurobiology to build more adaptive and robust robot controllers
(b) Passively compliant joints to dynamically scale the robot to match the model organism
(c) Biomimetic force and strain sensing to model sensory processing in the animal and improve the robustness of our robot's sensory system
(d) Passively compliant, tarsus-like foot segments to model insect foot-ground interactions.

By combining all of these features into a single robot, we aim to provide a comprehensive platform for experimentation regarding the neural mechanisms of the insect.

As such, we present Drosophibot, a morphological, hexapod robot inspired by the common fruit fly, *Drosophila melanogaster*, and designed to leverage broadly applicable discoveries from a variety of insects. We use what data we have from other insects when the data is not available in the fruit fly. This leads us to generate hypothetical models that can be tested in the fruit fly. In this manuscript, we begin by describing the conceptual design of the robot, such as its anticipated scale, actuation, and sensing. We then present a neuromechanical simulation of the robot and show that the conceptual design features replicate the biomimetic functions described above. Next, we describe a dynamical neural controller that controls the simulation's walking. We then describe Drosophibot's hardware and validate that the robot can replicate many key biomimetic features of the neuromechanical simulation. Finally, we discuss how

to increase the similarity between the simulation and the robot and how we plan to use Drosophibot in the future.

2. Conceptual robot design

Developing a morphological robotic platform capable of fulfilling our stated goals involves careful consideration of biological fidelity and mechanical functionality. To be useful for biological science, the robot needs to include a high degree of bio-mimicry in its components. However, too much fidelity might head to complexity that hinders the function of the robot as a mechanical platform. To maintain a reasonable balance between these two considerations, we made a series of conceptual decisions regarding Drosophibot's design. This section details these choices, as well as the justification for their inclusion.

2.1. Mechanical overview

For ease of manufacturing, assembly, and maintenance, we decided to make the robot 100 times bigger than *Drosophila*, resulting in a leg segment length of 10 cm. From there, Drosophibot needed to possess six legs. We elected to include three actuated joints within each leg. The thorax-coxa (ThC) joint protracts (+) and retracts (−) the leg, the coxa-trochanter (CTr) joint elevates (+) and depresses (−) the leg, and the femur-tibia (FTi) joint extends and flexes the tibia segment. Figure 1 shows each of these DOF on a leg of the robot. Each leg also has several tarsal segments passively actuated by a single elastic tendon. One additional actuated degree of freedom rotates the head in the yaw direction, totaling 19 actuated and 6 passive DOF throughout the robot.

The three active DOF per limb represent a simplification of insect leg anatomy. In reality, the ThC joint possesses multiple axes of rotation. However, multiple actuators would increase the robot's weight, jeopardizing its ability to walk. Therefore, robots are commonly designed such that the ThC axes are combined into one effective axis [20, 26, 42, 74]. Additionally, the precise DOF in *Drosophila's* legs have not been identified in detail. When they are, we will revisit the robot's leg design. Currently, the simplified DOF chosen for Drosophibot's limbs provide the necessary movement to support a high biological fidelity controller while minimizing overall robot weight, compromising sufficiently between functionality and bio-mimicry.

2.2. Actuation strategy

To minimize the mechanical and control complexity for the robot, we elected to actuate Drosophibot's limbs with DC servomotors. Servomotors require less specialized infrastructure (i.e. no compressor) to operate compared to pneumatic or hydraulic

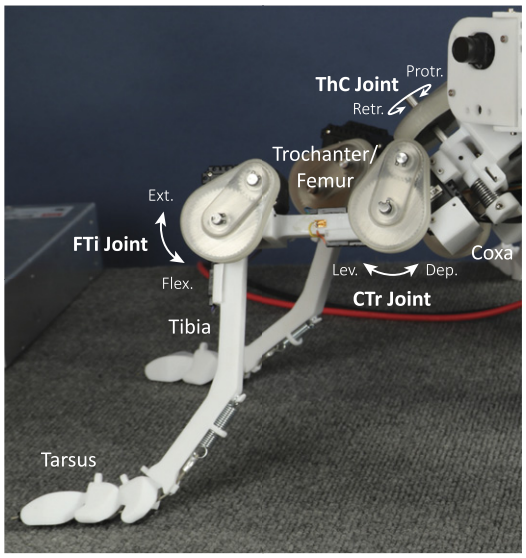


Figure 1. A view of one of the legs of Drosophibot with the anatomical components and DOF labeled. Each degree of freedom also includes arrows showing the directions of movement and their corresponding terms.

actuation, simplifying the mechanical and control design. However, these considerations are inconsequential if the servomotors can not accurately mimic the actuation present in the insect. Previous biological experiments suggest that insect joints have some functional similarities to servomotors, for example, in the firing frequency of the motor neurons (MNs) that innervate antagonistic muscles determines the steady state rotation of the joint [1, 33]. The stiffness of the muscles and membranes that span the exterior of the joints also provide a restoring force akin to the proportional negative feedback of a servomotor. When dynamically scaled, the damping ratio and natural frequency of a servomotor system with the selected parallel elasticity described in the next section is similar to those of a leg joint in *Drosophila* [65]. Section 5.1 has performance data of the robot's joints when combined with aspects of the robot's controller. With this idea established, we were able to utilize the mechanical and control simplicity of servomotors in Drosophibot's design while maintaining biological mimicry.

2.3. Passively compliant joints and dynamic scaling

A primary goal in the design of Drosophibot was the inclusion of animal-like passive compliance in the robot's joints. Joint compliance can be categorized as either in series or in parallel with the actuator. For example, a tendon or apodeme would be a source of series elasticity for a muscle, while skin or joint-crossing cuticular membranes would be a source of parallel elasticity. Robots such as HECTOR, AMOS II, and Octavio have previously incorporated series elasticity. HECTOR's elasticity is provided by integrated SEAs with the elastic component embedded in a

custom design [20]. Octavio and AMOS II also include series elastic elements in the form of couplings and spring clutches external to the actuator [26, 74]. Series elasticity has the primary benefit of reducing the rate at which force can act on the actuator. Additionally, measuring the deflection of the elastic element can serve as a proxy for actuator force.

In addition to series elasticity, parallel elasticity in animals works to provide a restoring force to the joint upon movement. For insects, such force serves a crucial role in joint behavior. Insects have a gravity-independent resting posture, even when their muscles are relaxed or deafferented [33]. Insect joints even exhibit a restoring torque when the muscles have been removed from the skeleton [33]. Stimulating muscle fibers causes only transient joint rotations; once muscle fibers relax, the limb returns to its original configuration [1]. Simulation studies show that for EMG data from muscles to correctly drive a simulated limb, both elastic and viscous forces that resist muscle contractions must be included [82]. These behaviors may be explained by how material properties should change with scale: Mass is proportional to length-scale cubed, but spring stiffness is proportional to length [38]. Therefore, small animals' body mechanics are dominated by elastic rather than inertial forces. However, we desire the robot to behave like an insect rather than an animal of the robot's size.

To align with our primary goal of facilitating the development of dynamic, morphological control networks, we designed our robot's joints to have an insect-like balance between elastic and inertial forces. The first design choice we made to achieve this joint elasticity was placing torsion springs in parallel with every leg joint. The springs are positioned such that their equilibrium positions are biologically plausible. The desired stiffness of these springs are such that the limbs can exhibit insect-like lower level behaviors (gravity independent posture, passive return reflex, etc) while still allowing the actuators to move against the springs without sustaining long-term damage. Such a stiffness greatly differs from the base values present in the animal because the robot is so much larger, so we additionally wanted to tune these elements such that the dynamic scaling of the robot is the same order of magnitude of the insect. By dynamically scaling a robot to match its model organism, we ensure that the timing of its motions are scaled to its mechanics. Many studies have used the Froude number (the ratio between kinetic and gravitational potential energy) as a way to enforce dynamic similarity between animals and models [8]. However, since a fruit fly's body posture is dominated by elastic forces rather than gravitational forces, we used the Strouhal number (the ratio between kinetic and elastic potential energy) to enforce dynamic similarity between the animal and our model. This approach has been used in related studies of locomotion [39]. A robot that models this behavior can have limbs of any size or mass and joints of any elasticity, as long as it steps at the same frequency, relative to its natural frequency, as the animal.

To dynamically scale the motions of Drosophibot to that of *Drosophila*, we computed the Strouhal number st squared,

$$(\mathrm{st})^2 \equiv \frac{T_{\mathrm{cycle}}^2}{T_{\mathrm{n}}^2} = \frac{1}{4\pi^2} \cdot \frac{k_{\mathrm{joint}}}{J_{\mathrm{limb}}} \cdot T_{\mathrm{cycle}}^2, \qquad (1)$$

where T_{n} is the natural period of the limb segment, which has units of time and is a function of the joint's torsional stiffness k_{joint} and the limb segment's moment of inertia about the joint J_{limb}. Additionally, T_{cycle} is the duration of one stepping cycle with units of time, meaning that $(\mathrm{st})^2$ is dimensionless. The value of $(\mathrm{st})^2$ must be the same for both Drosophibot and *Drosophila* to ensure that both operate in the same dynamic regime. If $(\mathrm{st})^2 > 1$, then $T_{\mathrm{cycle}} > T_{\mathrm{n}}$ and the robot and animal operate in an elasticity-dominated regime, sometimes called quasi-static [58]. If $(\mathrm{st})^2 < 1$, then $T_{\mathrm{cycle}} < T_{\mathrm{n}}$ and the animal and robot operate in an inertia-dominated regime. Direct measurements of k_{joint} and J_{limb} are not available for the fly, but we used scaling laws [38] and mass measurements [61, 81] to approximate these values and obtain an order-of-magnitude approximation of how quickly or slowly Drosophibot must step to capture the dynamic scale of the fruit fly. These calculations are presented in appendix A. We calculated that with its present joint stiffness Drosophibot must walk with a stepping period of 1.6 s to be dynamically similar to *Drosophila* running at its highest speed. We designed Drosophibot's controller to walk with a period of 2 s. Thus, the robot has a similar dynamic scaling to the insect while walking, ensuring the passive compliance in the joints mimics that in the animal while still maintaining mechanical feasibility.

2.4. Force and strain sensing

Insect campaniform sensilla have long been known to measure the strain of the animal's limbs, working in conjunction with force sensors built into the muscle–apodeme complex to sense a leg's overall load state. The animal's nervous system uses this sensory feedback to determine when legs are in stance or swing phase, communicate the state of the legs, and counteract unexpected perturbations while walking [85]. HECTOR and MantisBot are, to our knowledge, the only two robots that directly measure the strain in their limbs at these biological locations [20, 63]. Many robots that do not measure strain in this way instead sense torque by using motor current draw as a proxy for torque or to measure the deflection of an elastic component in the drive train, as in SEAs [26, 74]. For a system without parallel joint elasticity, motor current draw may be a suitable analog for biological leg loading data. However, we had chosen to include parallel elasticity in Drosophibot's joints, meaning that the motor torque would not reflect the full

loading state; the deflection of parallel components alters the torque that acts on the actuator. Therefore, we chose to include strain gauges as analogs to campaniform sensilla on Drosophibot. With these sensors, we can measure overall leg strain as an analog for joint torque, providing biologically similar strain data to support the desired morphological control network.

2.5. Passively compliant, tarsus-like feet

In addition to compliance in their leg joints, insects have compliant feet. Compliant feet increase the overall ground contact area and lower the impact loading when a leg enters stance phase, making them valuable to include on robots to prolong actuator life and the load-bearing components. AMOS II and Octavio include some manner of passive compliance in their feet, through springs in the tip of the final leg segment or a dedicated foot segment with a passive ankle joint, respectively [26, 74]. Additionally, most insects have fairly long tarsi, with *Drosophila*'s tarsus measuring almost the same length as the tibia. Without such feet segments, insect-inspired robots' torsos sit much closer to the ground than those of insects, potentially causing unintended collisions of the torso with the ground and making it more difficult to compare the kinematics of the robot to the model organism. Longer, tarsus-like feet help rectify this mismatch, facilitating better tests of biological fidelity in controllers. Implementing compliance in these feet segments further increases bio mimicry while also providing beneficial impact force reduction. To help maintain similar leg proportions to the insect and prevent slip, we elected to design Drosophibot with compliant, segmented tarsi like those of *Drosophila* and other insects. Such feet will allow us to study biological foot-ground interactions, while also improving the ground grip of the mechanical system.

3. Simulation as a computational neuromechanical model of a fly-like robot

To test the basic mechanical constraints described in the previous section and to further design the robot, we used AnimatLab [12] to build a rigid-body simulation with a closed-loop neural controller. The simulation lets us test whether our actuator choice is reasonable, whether our specifications are practical, and whether the neural controller can produce walking behavior. In the following, we describe the simulation and what it tells us about the robot.

3.1. Mechanical simulation

Drosophibot's mechanics were simulated via a rigid-body simulation. Each segment was modeled as a rigid (i.e. unbending) rectangular prism of uniform mass (as measured from the robot), and each joint was modeled as a hinge with one rotational degree of freedom. Each leg possessed four DOF: the three actuated joints listed in section 2.1 and a passive tibia-tarsus (TiTar) joint. For simplicity, the tarsomere segments of each tarsus were modeled as one rigid segment. However, a spring and dashpot were added to each leg joint in order to mimic the passive dynamics of the robot. This includes the passive elasticity of the compliant tendon in the robot's tarsus. In this study, we did not thoroughly test the impact of the compliant tarsi on the robot's or simulation's performance. However, the tarsi's compliance anecdotally improved the simulated robot's ability to make even contact with the substrate versus when the tarsi were locked in place.

3.2. Neural controller assembly and tuning

Drosophibot's controller is constructed entirely of leaky integrator nonspiking neurons, using a system of models and design tools we call a 'synthetic nervous system' (SNS) [66, 67]. This technique is used to build dynamical, structurally-transparent, bio-constrained models of animal nervous systems that model and explain results from neurobiology while simultaneously applying neurobiological results directly to the field of robotics. SNS have been successfully applied to replicate both the function [64, 69] and neural activity [68] of the nervous system in robotic models of insects. SNS models may directly model the dynamics of nonspiking neurons that communicate via graded-potential nonspiking chemical synapses or approximate the average firing rate of a population of spiking neurons [78]. The model dynamics are explained in appendix B.

A portion of Drosophibot's leg controller is shown in figure 2. Its architecture is strongly based on our group's previous neural stepping controllers [55, 64, 68–70]. The network is hierarchically organized, with lower levels possessing as much autonomy as possible while still generating coordinated body-wide motion. We will explain the hierarchy from the top, down.

3.2.1. Descending influences

The highest part of the control hierarchy sets the forward progress and rotation of the body achieved by each step (figure 2(a)). These two parameters specify the motions that the feet, and therefore the leg joints, must execute throughout each step [64, 70]. These values are set by 'volitional commands,' which are applied tonic currents to these neurons. In insects, such inputs are expected to be the result of activity in the central complex (CX) of the brain [44].

The 'body translation' and 'body rotation' neurons synapse onto interneurons in each joint's local control network (figures 2(d) and (e)) to alter the sign of reflexes within each leg. Such alterations are thought to underlie walking in different directions [44]. These connections are tuned by computing inverse kinematics for walking in different directions,

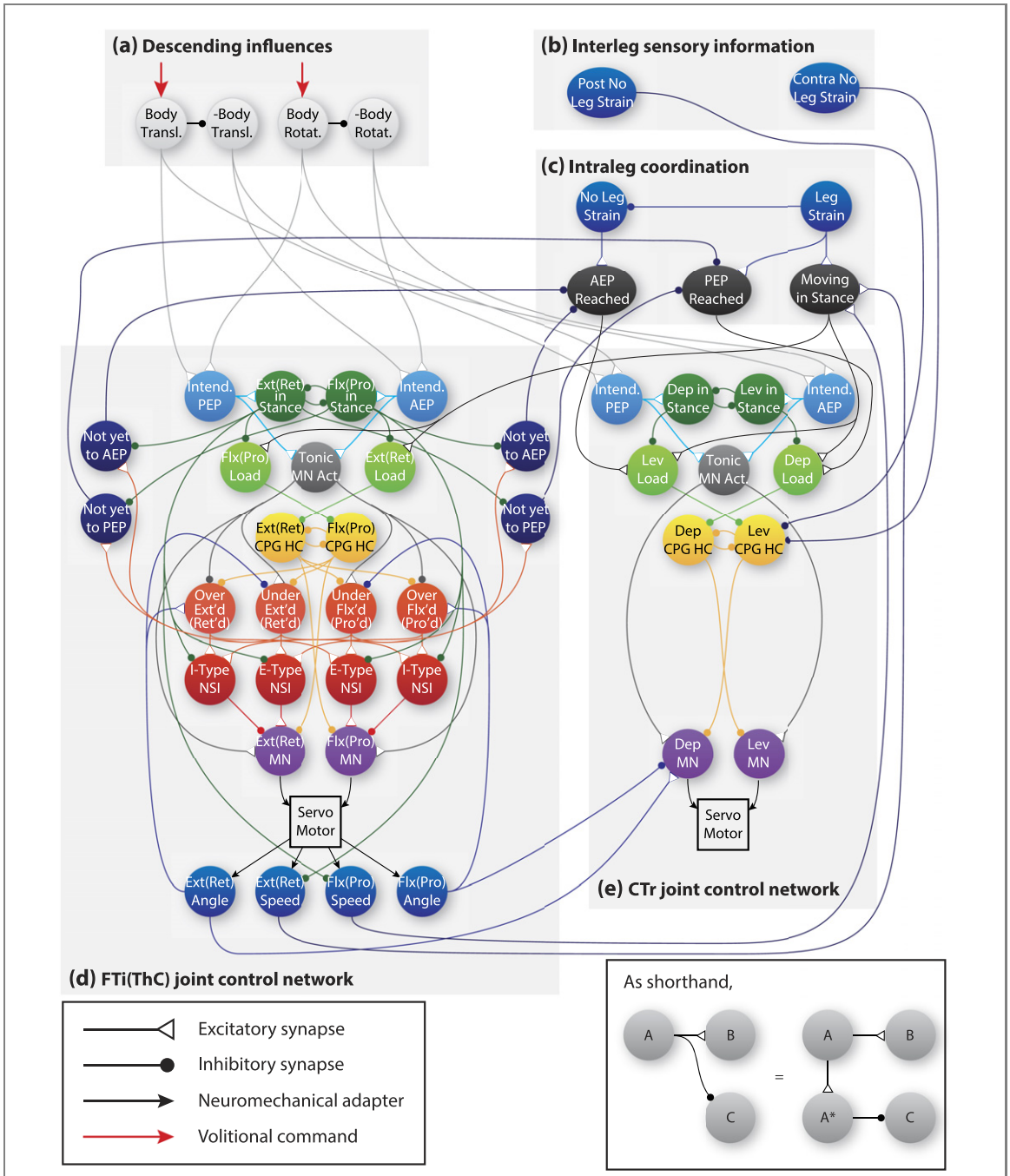


Figure 2. Diagram of the network architecture used to control the walking of Drosophibot in simulation. The network is hierarchical. (a) Sparse descending commands project throughout the leg control networks to encode the intended walking direction. (b) Sensory information from the contralateral leg and the posterior ipsilateral leg influence the timing of leg depression and levation. (c) Both proprioceptive and force feedback are integrated to create signals that entrain all of the joints in the leg. (d) The FTi and ThC joints have identically structured networks. (e) The CTr has a reduced network, because its function is simply to depress or levate the leg. Note that the network diagram has been simplified by removing inhibitory interneurons. Instead, some neurons are drawn as forming both excitatory and inhibitory connections. Acronyms: AEP, anterior extreme position; PEP, posterior extreme position; CPG, central pattern generator; HC, half-center; NSI, nonspiking interneuron; MN, motor neuron.

and then encoding correlations between body motion and joint rotation into the synaptic connections [70]. Each joint's network possesses an 'intended PEP' and 'intended AEP' neuron (light blue, figure 2(d)). These neurons feed into a bistable decision network that inhibits connections within the joint controller to ensure that it either flexes or extends during stance phase (dark green, figure 2(d)). The bistable network inhibits: Timing signals to one half of the pattern generating network, to control which motion takes place during stance phase (light green, figure 2(d)); nonspiking interneurons (NSIs) that control swing phase motion (red, figure 2(d)); and AEP and PEP signaling neurons (navy, figure 2(d)). More details about this process and tuning this network can be found in [64, 70].

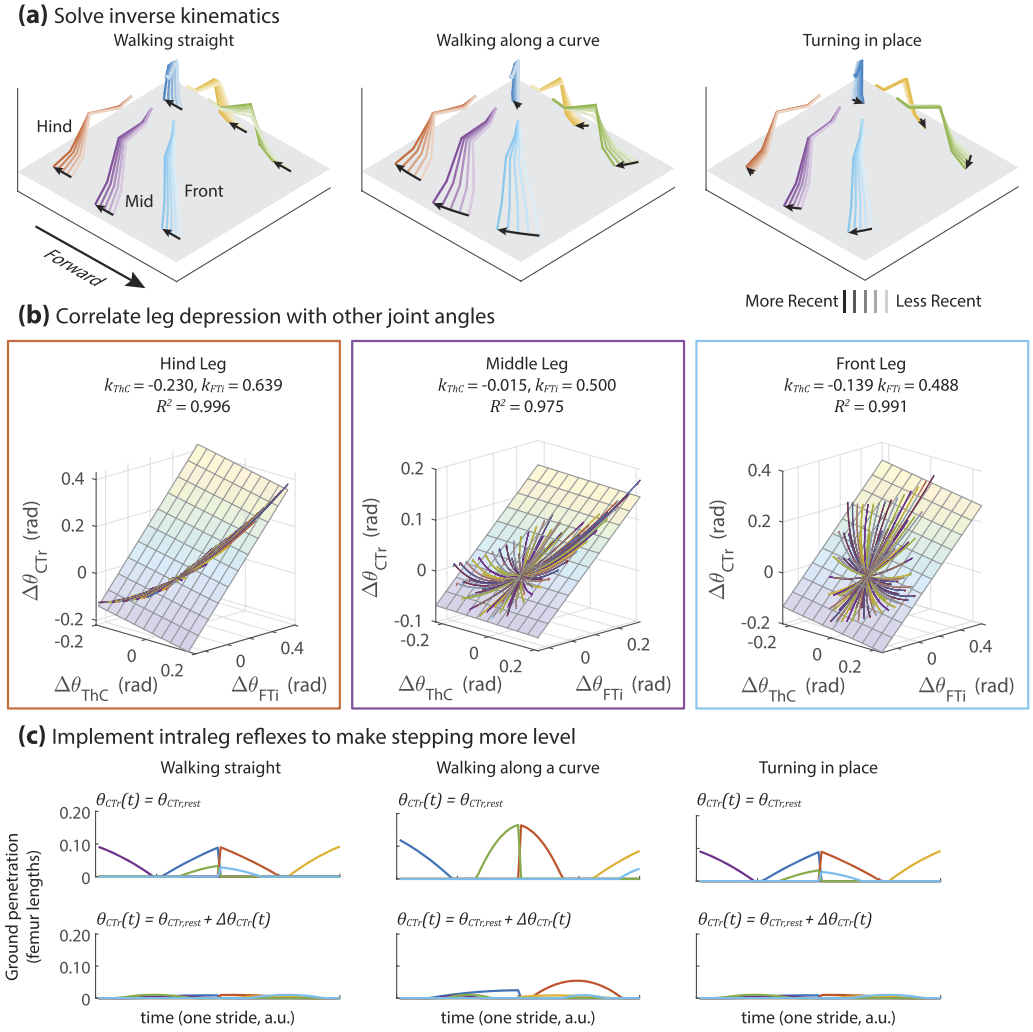


Figure 3. Figure illustrating the tuning and effect of intraleg reflexes that modulate leg depression. (a) To understand how CTr depression should be modulated to keep stepping height level, the inverse kinematics of stepping were solved for many different directions and stride lengths, three examples of which are illustrated. (b) It was found that CTr angle correlates very strongly with the other two leg angles throughout the stance phase for each leg (border color corresponds to leg color in (a)). This correlation was used to tune neural reflex pathways in the leg controller. (c) When the simulation simply depresses its CTr joints to a constant angle throughout the stance phase (top row), the foot penetrates the ground, which would lead to unsteady posture. However, including reflexes to adjust CTr depression as a function of the other joint angles (bottom row) produces much more level stepping.

3.2.2. Interleg coordination

Interleg timing is coordinated by simple connections between the legs based on the so-called Cruse rules [13] (figure 2(b)). We adapted connections from our previous work [55] to enforce the following coordination rules: (1) a leg may not enter swing phase when its posterior ipsilateral or contralateral leg is unloaded; and (2) a leg is temporarily encouraged to enter swing phase when its posterior ipsilateral leg enters stance phase.

3.2.3. Intraleg coordination

The timing of joint motions is coordinated into a stepping pattern by proprioceptive and strain feedback: a combination of load and velocity feedback excite the 'moving in stance' neuron (figure 2(c)), which reinforces the stance phase portions of the pattern generators (yellow, figures 2(d) and (e)). When this signal decays or the 'PEP reached' neuron fires, then the swing phase portions of the pattern generators are disinhibited. Swing phase ends once the pattern generators flip back into stance phase or when the 'AEP reached' neuron fires. In this way, the multiple pattern generators in the leg are coordinated into a stepping pattern only when sensory feedback is present [10].

The magnitude of CTr depression is regulated by proprioceptive reflexes from the ThC and FTi joints. In cockroaches [18] and stick insects [31], rotating the FTi joint is known to alter the firing rate of the depressor and levator MNs. Such reflex action is presumed to help the animal maintain a constant stepping height throughout the stance phase [18].

We applied our functional subnetwork approach [66] to tune the strength of intraleg reflexes that depress or levate the leg to maintain constant stepping height. First, for a given leg, we solved the inverse

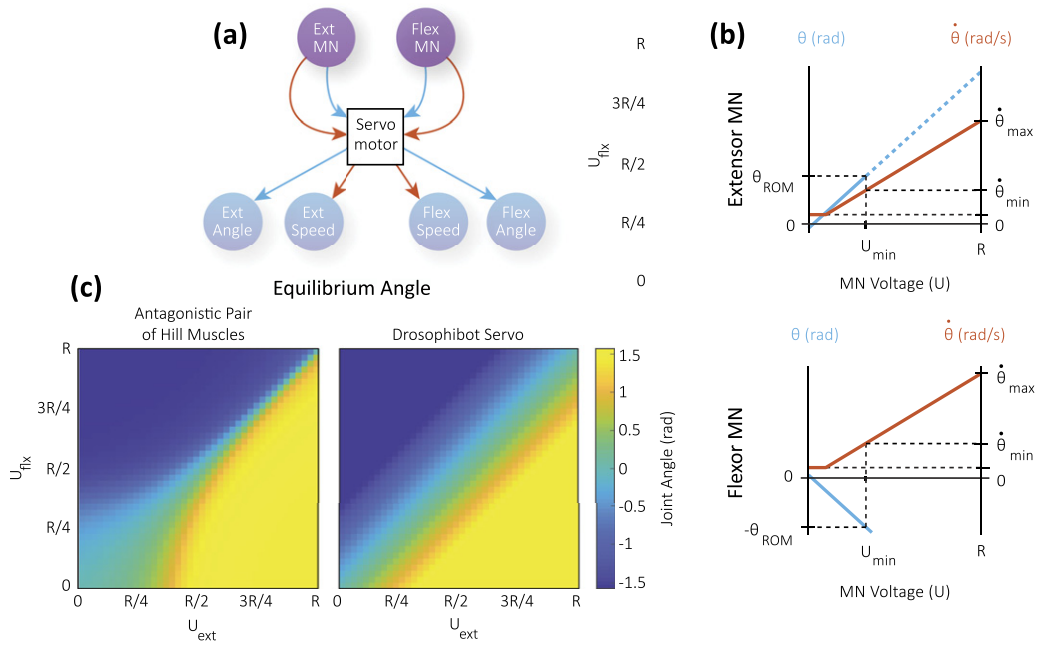


Figure 4. (a) Network diagram illustrating that each MN commands both an angle and movement speed to the servomotor. The servomotor interprets the sum of the commanded angles as its commanded angle, and the sum of the commanded speeds as its commanded speed. (b) Plots illustrating how the MN voltage maps to both a commanded angle and movement speed between zero and the MN pre-synaptic saturation voltage (R). (c) The servo control scheme mimics the net effect of a pair of antagonistic muscles. In both scenarios, equally activating the MNs results in no joint rotation, and activating one MN without the other causes the joint to rotate to its limit.

kinematics problem as that leg stepped in a spectrum of directions and amplitudes [64, 70] (figure 3(a)). Then, we plotted the CTr angle as a function of the ThC and FTi angles, from multiple trials (15 step directions and 7 step amplitudes) (figure 3(b)). No matter the stepping direction or amplitude, the CTr angle could be described as a planar function of the other two angles. Therefore, we could use the proprioceptive mappings from section 3.2.5 and the motor output mappings from section 4 to find the effective coupling k between one joint angle and the CTr joint angle.

$$\theta_{\mathrm{CTr}}(t) = \theta_{\mathrm{CTr,rest}} + \Delta\theta_{\mathrm{CTr}}(t), \tag{2}$$

where

$$\Delta\theta_{\mathrm{CTr}}(t) = k_{\mathrm{ThC}} \cdot \left(\theta_{\mathrm{ThC}}(t) - \theta_{\mathrm{ThC,rest}}\right) + k_{\mathrm{FTi}} \cdot \left(\theta_{\mathrm{FTi}}(t) - \theta_{\mathrm{FTi,rest}}\right). \tag{3}$$

To test that this intraleg reflex can produce level stepping, we computed the forward kinematics for each tarsus while the body was held at a constant height. This enabled us to compare the positions relative to where the ground would be, i.e. the ground plane. Figure 3(c) shows the effect of this reflex on the foot depression. If $\theta_{\mathrm{CTr}}(t) = \theta_{\mathrm{CTr,rest}}$, then the foot may substantially penetrate the ground plane (up to 20% of the length of the femur). This was a challenge identified in our previous work [70]. However, if $\theta_{\mathrm{CTr}}(t) = \theta_{\mathrm{CTr,rest}} + \Delta\theta_{\mathrm{CTr}}(t)$, then these problems are greatly ameliorated. Therefore, the sensory neurons that encode angle directly synapse onto the depressor MN of each leg to maintain level stepping (navy, figures 2(d) and (e)).

3.2.4. Actuation and motor output

The lowest part of the hierarchy controls both the timing and amplitude of motion of the joint [11]. The network interfaces with servomotors via an antagonistic pair of slow, excitatory MNs (violet, figures 2(d), (e) and 4(a)). To explain how the network and actuators interface, let us first explain how the actuators are controlled. Each joint is actuated by a Dynamixel smart servo (Robotis, Seoul, South Korea). The servos are 'smart' because each possesses its own ARM-based microcontroller, which performs low-level feedback control of the servo's position. As such, the user controls the rotation of a servo by commanding an angle to the network address associated with that servo. Additionally, the user can limit (but not directly control) the speed at which the servo rotates. This speed command is not direction-specific, and is therefore truly a speed (not velocity) command.

To exploit this control scheme, each MN's activation is mapped to both a commanded angle and a commanded maximum speed for the servomotor. How should these values be mapped? Assume that during walking, a servo must sweep its range of motion θ_{ROM} regardless of walking speed. When the robot walks at its maximum speed, the joint sweeps θ_{ROM} over some period of time T_{min}. When the robot walks at its minimum speed, the joint sweeps θ_{ROM}

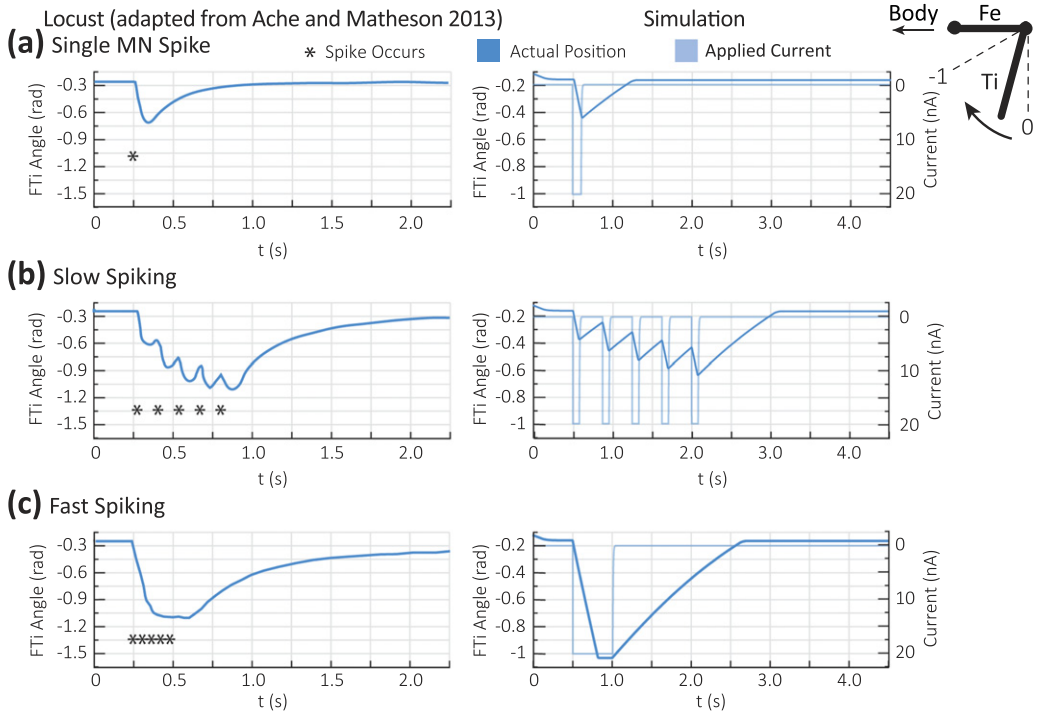


Figure 5. The simulated robot's joint response to MN stimulus mimics that of the locust. Data from the locust (left column) is adapted from the mean joint response depicted in figure 1 of [1]. (a) The model's joint's response to a brief command to flex mimics the animal's joints' response to a single MN action potential, i.e. brief, rapid flexion followed by a longer, passive extension to equilibrium. (b) The model's joint's response to sparsely administered, brief commands to flex mimics the animal's joint's response to sparse MN action potentials, i.e. an accumulation of flexion followed by passive extension to equilibrium. (c) The model's joint's response to a persistently administered command to flex mimics the animal's joint's response to dense MN action potentials, i.e. smooth, rapid flexion followed by passive extension to equilibrium.

over some period of time $T_{\max} = n \cdot T_{\min}$. Thus, the minimum and maximum servo angular speeds are

$$\dot{\theta}_{\max} = \frac{\theta_{\text{ROM}}}{T_{\min}} \tag{4}$$

$$\dot{\theta}_{\min} = \frac{\theta_{\text{ROM}}}{T_{\max}} = \frac{\theta_{\text{ROM}}}{n \cdot T_{\min}}, \tag{5}$$

where n is the ratio between the maximum and minimum speeds.

We wish to encode the joint speed in the voltage of the MN above its resting potential, U (figure 4(a), see also appendix B), because the angular speed of leg joints in the cockroach correlate to MN activity [75–77]. When $U = R$, the maximum value for U, $\dot{\theta} = \dot{\theta}_{\max}$. It follows from equation (5) that when $U = n \cdot R$, $\dot{\theta} = \dot{\theta}_{\min}$. Thus, we can establish a linear encoding of the joint's commanded angular speed $\dot{\theta}$ in the MN activity U (figure 4(b), red line).

We also wish to encode the joint angle in the voltage of the MN, because the tension of antagonistic muscles establishes the equilibrium angle of an animal's joint [65]. As established above, the joint angle must sweep the angle θ_{ROM} no matter the angular speed. Therefore, we require that $\theta \geqslant \theta_{\text{ROM}}$ for all $U \geqslant U_{\min} = n \cdot R$. In the cases where $\theta > \theta_{\text{ROM}}$, we assume that intraleg coordinating influences will cause the joint's CPG to 'flip' and reverse the joint's direction once $\theta \approx \theta_{\text{ROM}}$. Thus, the joint's commanded equilibrium angle is a linear function of the MN activity U that includes the point $\theta(U = U_{\min}) = \theta_{\text{ROM}}$ (figure 4(b), blue dashed line). Note that antagonistic MNs encode the position with opposite signs. The command sent to the servo is the sum of these two values.

Figure 4(c) shows that the servo's equilibrium joint angle depends on the activation of antagonistic MNs. The resulting relationship is qualitatively similar to that of an antagonistic pair of muscles. Specifically, activating either antagonist too strongly causes the joint to rotate to one extreme, but activating them to similar levels causes the joint to stay in the middle of its range of motion.

This actuation scheme of using MN voltage to command a servomotor's position and speed produces a dynamic response similar to that observed in locusts [1]. Figure 5 plots the mean joint angle time course for experiments from [1] alongside data from similar experiments performed with the simulated robot joints. Briefly commanding the simulated joint to flex causes the joint to flex rapidly and then slowly extend back to its equilibrium position. This response is similar to the locust's joint rotation after the slow flexor MN fires a single action potential (figure 5(a)). Repeatedly administering brief commands for the simulated joint to flex results in an accumulation of joint flexion over time, as observed in the locust leg joint (figure 5(b)). Administering a persistent command for the simulated joint to flex

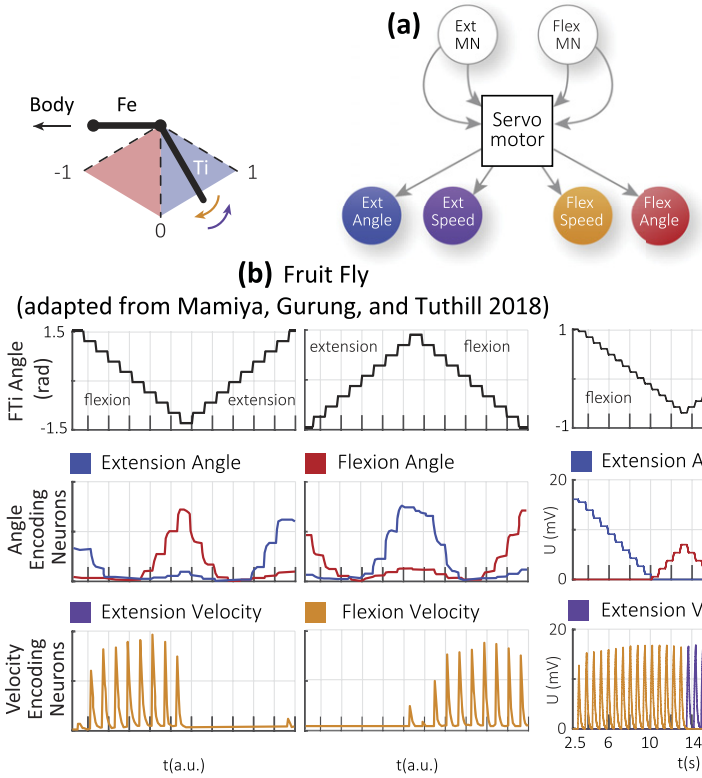


Figure 6. (a) Each servomotor returns proprioceptive feedback to four neurons that encode distinct quantities: the amplitude of angle extension past the neutral angle (blue); the extension speed (purple); the flexion speed (orange); and the amplitude of angle flexion past the neutral angle (red). (b) Data adapted from figure 5 of [43] showing the average GCaMP6f fluorescence of the groups of sensory neurons in response to stimuli. Separate populations encode the extension and flexion angle of the joint. Additionally, one population encodes the flexion velocity of the joint. (c) The analogous simulated neurons encode the same quantities. We inferred the existence of a population to encode extension velocity based on other data presented in [43].

results in smooth flexion followed by slow extension back to equilibrium. This response is similar to when the locust's flexor MN fires many action potentials in quick succession (figure 5(c)). The similarity in joint responses between the robot simulation and an insect justifies our choice of actuators and supports the biomimicry of the model.

In the full walking network, the MNs receive a tonic drive when the robot is in walking mode (gray, figure 2(d)), and are rhythmically inhibited by pattern generating neurons to produce alternating motion required for walking (yellow, figure 2(d)) [9]. In the network, the amplitude of this tonic drive encodes the intended angular excursion of the joint during a single step. These components form the basis of the single joint controller.

To more precisely control the motion of the joint throughout the swing phase, the network also includes a number of NSIs that are arranged in parallel antagonistic pathways [79]. These NSIs integrate sensory feedback from both extension and flexion sensors to calculate both excitatory and inhibitory adjustments to MN activity during the swing phase. The upper layer of NSIs (orange, figure 2(d)) compare the measured joint angle to the intended angle, and are rhythmically inhibited by the CPG. The lower layer (red, figure 2(d)) combine the sensory-driven responses of the upper layer across the two 'halves' of the network (e.g. flexion and extension) to compute corrections to MN activity [57]. The NSIs that would excite or inhibit the stance phase MN are inhibited depending on how that joint moves (e.g. whether if flexes or extends during stance phase). This assembly of NSIs does not represent the full diversity of NSI types or all of the functions they may perform. However, these aid in controlling the amplitude of the swing phase motion.

3.2.5. Sensory feedback

The network receives sensory feedback from proprioceptors and strain sensors. Proprioceptive information is encoded by four classes of sensory neurons, which respond separately to joint flexion, joint extension, flexion speed, and extension speed (figure 6(a)). These reflect four of the identified afferent neuron types of the CO in *Drosophila* [43]. Figure 6(b) plots the mean sensory neuron response (i.e. instantaneous firing frequency normalized to the maximum firing frequency) of proprioceptive populations in *Drosophila* as the tibia is rotated [43]. Each proprioceptive population responds to a different feature of the motion. Figure 6(c) plots the response of analogous neurons in the simulation. The animal afferent responses are more complicated than those

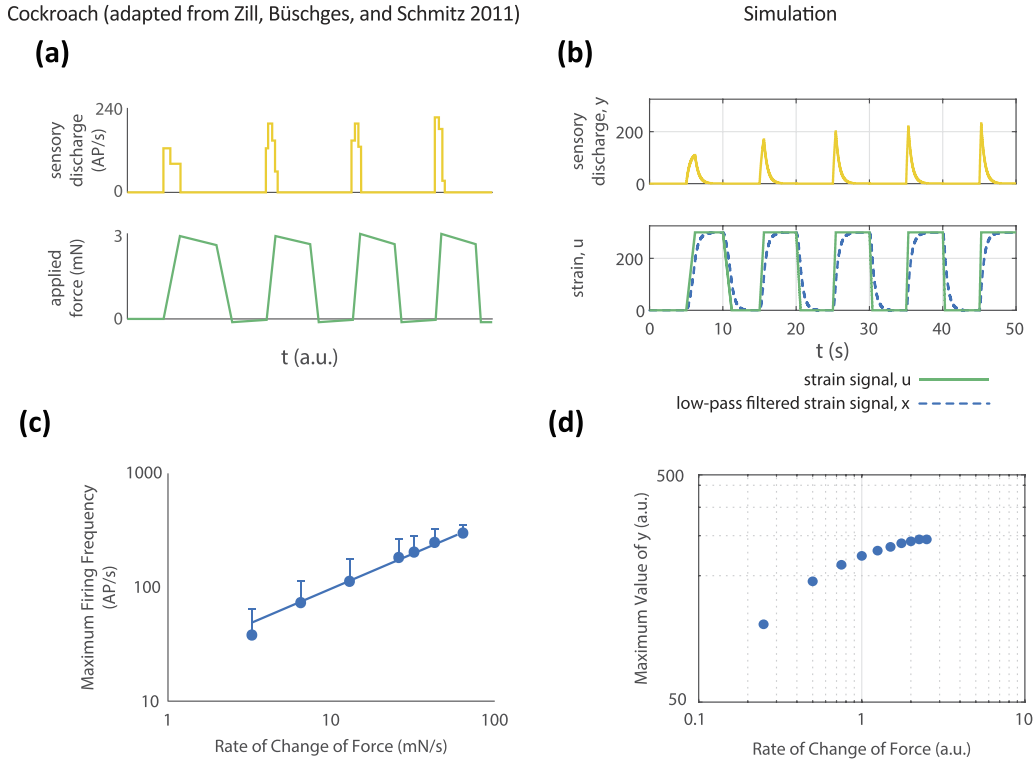


Figure 7. Comparison between the response of group 6B mid tibial campaniform sensilla in cockroaches (a) to the simulation (b). The response of both the animal's afferents (c) and the simulation (d) monotonically encode the rate of loading on the sensor.

of the neurons we included, namely in that they are highly dynamic and exhibit hysteresis [43]. It is not presently clear what role these dynamic responses play in the animal, so our model does not include them. We instead use the servo's internal potentiometer to directly measure its angle, meaning that there is no hysteresis or adaptation in its response. If the dynamic behaviors of these afferents are shown to play a critical role in the control of walking behavior in the future, we will attempt to model these dynamics and better understand how they contribute to control. Presently, limiting the network to these forms of proprioception constrains the resulting hypotheses regarding motor control.

The walking control network uses proprioception for several purposes. The flexion and extension signals are compared to the network's intended flexion and extension (orange, figure 2(d)) to guide swing phase via NSIs [79] and compute if the leg has reached the anterior extreme position (AEP) to trigger leg depression or the posterior extreme position (PEP) to trigger leg levation (navy, figure 2(d)). The speed signals feed back onto the CPGs within the leg to reinforce ongoing motions (e.g. the 'active reaction' observed in stick insects [4]; see the 'moving in stance' neuron, figure 2(c), black [67, 69]).

The network also receives strain feedback from sensors on the leg to detect when the leg is supporting the body during stance phase. Feedback from strain sensors are known to critically affect the coordination of locomotion patterns in insects [85]. In the simulation,this strain was simulated as the passive deflection of the tarsal segment, which was spring-loaded and pretensioned like Drosophibot's foot. This bending signal is processed to transduce strain into neural activity as seen in campaniform sensilla. Specifically, afferent responses to bending the cuticle may reflect the amplitude of bending, the rate of bending, or both [83, 84]. We simulate this phenomenon as a filter in which the sensory response y is the instantaneous measurement u minus a low-pass filtered history of the measurement x. Such an operation causes the sensor to adapt to bending measurements over long time periods and leads to a rate-based encoding of bending. The filter uses the time constant τ and the time-step duration Δt to update the filter,

$$\begin{aligned} x_i &= x_{i-1} + \frac{\Delta t}{\tau} \cdot (-x_{i-1} + u_i) \\ y_i &= u_i - x_i. \end{aligned} \tag{6}$$

To test how well this model captures the processing of campaniform sensilla afferents, we ran a series of trapezoidal force time courses through the filter. Each subsequent trapezoid has a higher rate of increase and decrease. Figure 7(a) shows data adapted from [83] in which this experiment was performed on cockroach tibial campaniform sensilla. As seen in the animal, the model afferents are active during the force increase phase (figure 7(b)). In addition, the peak afferent activity monotonically encodes the rate of force application (figures 7(c) and (d)).

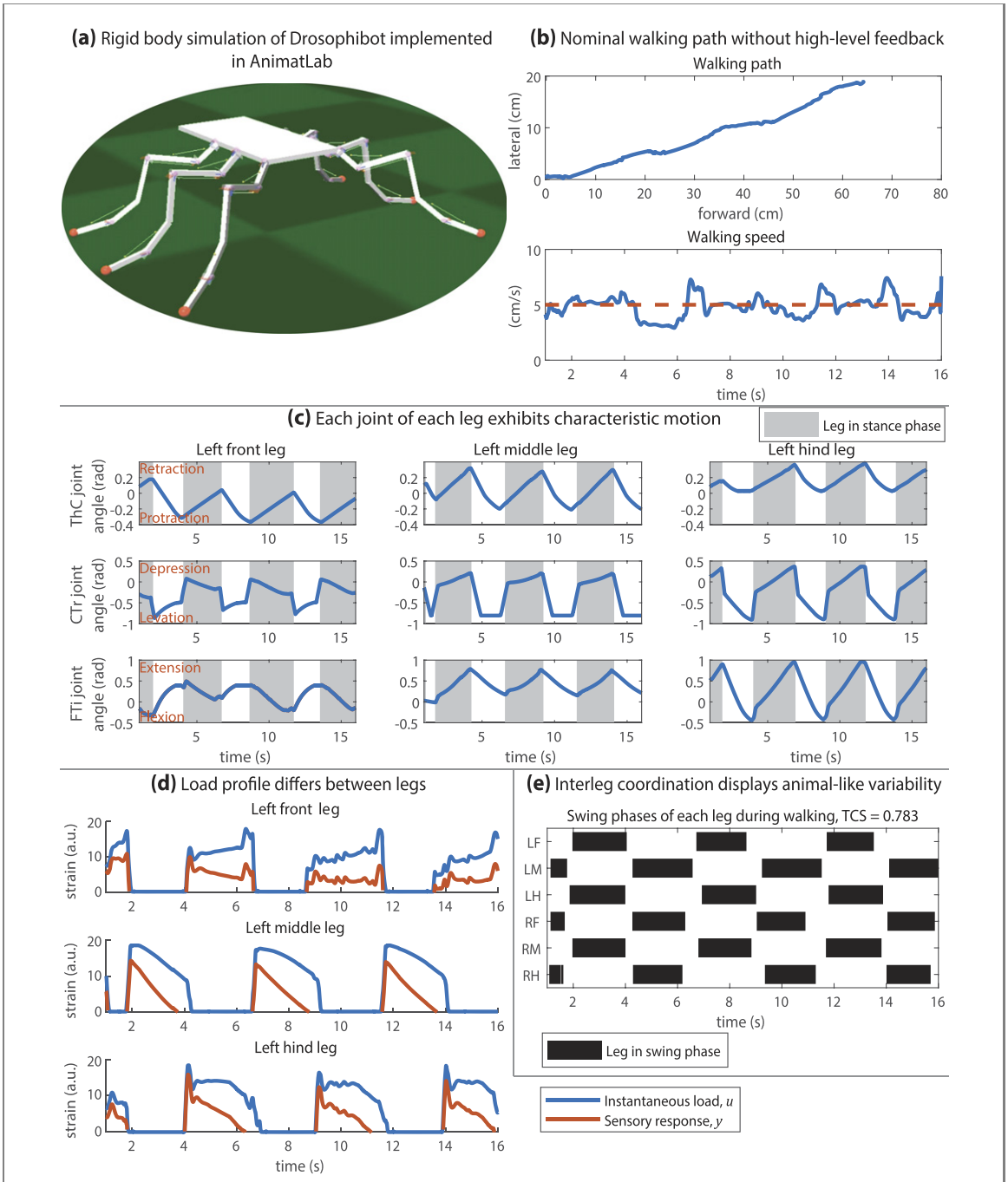


Figure 8. Data from a dynamic simulation of Drosophibot walking. (a) A screen capture of the rigid body simulation. (b) The simulation walks forward at the intended speed (dashed red line), despite lacking higher-level feedback to control the course. (c) Each joint in each leg performs a functionally-unique motion. (d) The instantaneous leg load u (blue) and corresponding sensory response y (red, see also equation (6)) varies in a leg-specific manner. (e) Interleg swing timing is coordinated, but shows variability consistent with the animal model.

This encoding scheme has the additional benefit of removing any constant bias in the force signal, enabling the sensor to self-calibrate.

3.3. Walking in simulation

To demonstrate that the presented subsystems and the neural controller can produce walking behavior, we collected data from a rigid-body simulation of Drosophibot while it walked (figure 8(a), video available here: https://youtu.be/lsiT8fcWf7o). The simulation was commanded via the 'volitional commands' in figure 2 to walk in a straight path at maximum speed. Recall that the network does not include any exteroception with which to measure heading or speed; the data in figure 8(b) is open-loop dead-reckoning. In spite of this, the simulated robot walks forward at the intended speed. Recall that this speed was chosen to make the motion dynamically similar to a fruit fly walking at its highest speeds (about 30 mm s^{-1} [81]).

As the simulated robot walks, each leg performs unique motions specific to that leg (figure 8(c)). This is the result of the network tuning, which encodes the strength and direction of the sensory reflexes that

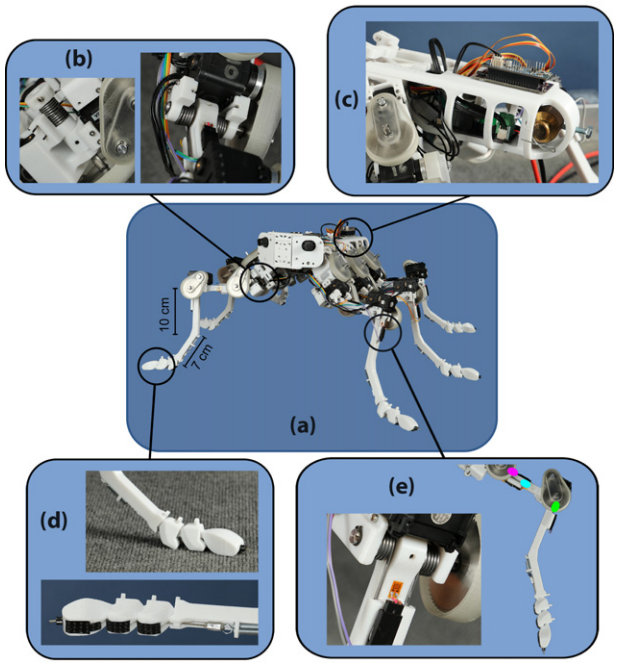


Figure 9. (a) A full body view of Drosophibot, a hexapod robot based on *Drosophila melanogaster* (b) views of the ThC joint (left) and CTr joint (right) torsion springs, which mimic the parallel elasticity in the insect (c) Drosophibot's abdominal segment, which houses electronics and can be adjusted to shift x-coordinate of the robot's COM (d) the robot's compliant tarsi. The rubbery, cleated feet passively deform for maximum ground contact (e) an example of one of the strain gauges on the FTi joint (left), as well as the locations of all of the strain gauges on the limb (right).

coordinate walking [70]. The robot's front legs reach in front of the body, extending the FTi joint during swing phase, and flexing during stance phase. This is in contrast to the middle and hind legs, which flex the FTi during swing phase and extend during stance phase [62, 71]. All of the legs retract the ThC joint during stance phase. All of the legs depress the CTr joint to begin stance phase, but the CTr may either levate or depress during stance phase to maintain a level step height. In spite of level stepping height, figure 8(d) shows that the legs' loading profiles are not identical. In particular, the front leg's load peaks at the end of stance phase, while the middle and hind leg's load peaks near the beginning of stance phase, as observed in walking stick insects [15]. Such functional differentiation was not explicitly designed into the system.

The variability of the relative timing of leg stepping in the simulation aligns with what is observed in fruit flies. The 'tripod coordination strength' (TCS) metric from [81] measures the degree to which the swing phases of one alternating tripod align. The higher the TCS, the tighter the interleg stepping timing. Even when walking at their highest speeds (i.e. with the tightest coordination), fruit flies rarely exhibit a TCS above 0.8 [81]. Figure 8(e) shows that the simulated Drosophibot walks with an average TCS of 0.783. This variability was not explicitly designed into the system.

4. Robot hardware description

After verifying the performance of the robot design in simulation, we constructed the hardware robot Drosophibot, shown in figure 9. In this section we describe unique mechanical and electronics features. In the subsequent section, we present results describing how well Drosophibot captures the animal-like responses modeled in the previous section.

4.1. Strain gauges

To mimic insects' load sensing abilities, each leg has three strain gauges to detect the strain of each leg during locomotion. Figure 9(e) shows an example of one of the strain gauges on the tibia, as well as the locations for all of the gauges on the limb. The magenta and cyan sections represent the anterior group 2 and dorsal groups 3 and 4 sensors on the trochantofemur, respectively [86]. The green section represents the dorsal group 6 on the tibia [83]. Load signals from the strain gauges are amplified by custom Wheatstone bridges, which then interface with the robot's control board. Section 4.5 provides more details on the electronic layout of the robot.

4.2. Joint compliance

Drosophibot's joints include torsional springs that produce insect-like passive elastic forces [33]. Figures 9(b) and (e) show examples of the springs on

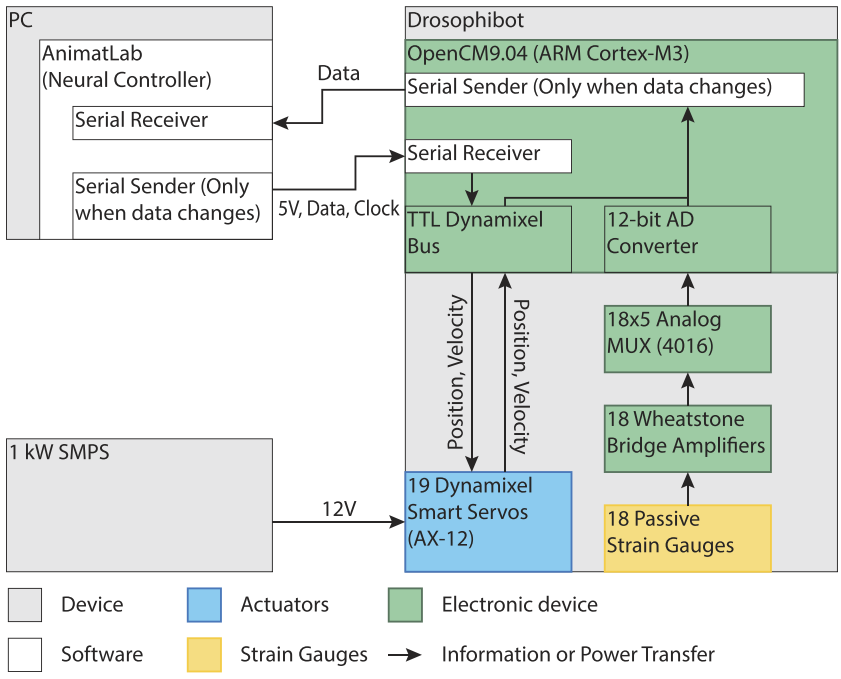


Figure 10. Diagram illustrating the power and control systems of Drosophibot. A desktop computer runs the neural simulation. The computer converts neural states into robot states and transmits these over a serial connection to the robot. The robot distributes actuator commands to the servomotors, and collects sensory data from both the servomotors and 18 strain gauges mounted on the legs. The robot returns this data to the desktop computer to complete the feedback loop, which runs at 50 Hz. The robot has its own 1 kW power supply.

each joint of the robot. The springs are positioned such that their intended rotational direction helps support the weight of the robot during stance while still maintaining biologically plausible equilibrium positions. In the ThC and FTi joints, the springs are completely secured to provide joint stiffness in both directions.

4.3. Tarsus-like feet

Figure 9(d) shows detailed views of Drosophibot's compliant, segmented tarsi. Each tarsus is comprised of an angled extension of the tibia with a distal series of three tarsomere segments interconnected with limited ball-and-socket joints that permit motion in the dorsoventral direction. Each segment is homologous save for the final, which is longer and wider for greater ground contact. The underside of each segment includes cleated rubber pads to reduce slip on a variety of surfaces. Cables route through the underside of each tarsus and attach to an extension spring mounted on the tibia, which provides a restoring force throughout stance phase of the limb. The stiffness of this spring was tuned to begin foot deflection under a third of the weight of the robot.

4.4. Center of mass

To mimic *Drosophila*'s distribution of mass, Drosophibot has an adjustable abdominal segment that houses its electronics. The center of mass (COM) of *D. melanogaster* is approximately 1/10 of the length of the thorax behind the middle set of legs [61]. However, Drosophibot's actuators heavily bias its COM to directly between the middle legs. Therefore, ballast can be inserted into the abdomen and its length can be adjusted to match the fore-aft component of the insect's COM (figure 9(c)). 200 g of additional mass adjusts the COM to a biologically plausible position.

4.5. Electronic hardware

Figure 10 summarizes the power and control systems on board Drosophibot. Drosophibot's 19 actuated DOF are actuated by Dynamixel AX-12 smart servos (Robotis, Seoul, South Korea), which have a stall torque of 1.5 N m at 11.1 V. Each leg servo includes additional gearing to double the output torque for needed strength. Drosophibot's 19 Dynamixel smart servos are powered by an off-board 12 V, 1000 W switching-mode power supply. Each leg is provided power in parallel to the others. To reduce voltage fluctuations on the main power rail, each leg possesses a 35 V, 3300 μF capacitor in parallel with the smart servos.

An OpenCM 9.04 microcontroller (Robotis, Seoul, South Korea) manages data transfer between the robot and the control computer. The control computer is an offboard laptop that runs the neural controller in the neural simulator AnimatLab [12]. We used the AnimatLab Robotics Tookit [63] to write a protocol for serial communication between AnimatLab and Drosophibot, named Szerial. Szerial uses user-defined maps to convert floating-point neural states into integer robot commands (i.e.

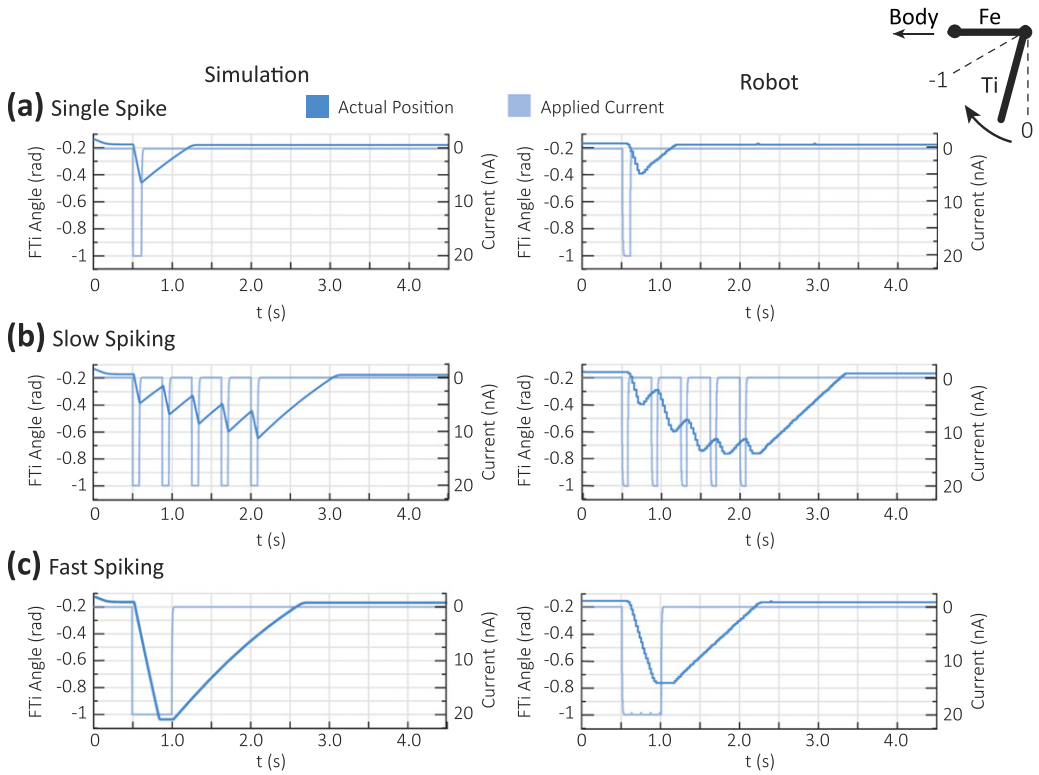


Figure 11. By adding in parallel elastic elements and rewriting the low level control of the servomotors, we are able to achieve similar passive elastic return as in the insect. The actual and commanded joint position of an FTi joint on Drosophibot in simulation and on the robot for a variety of flexion MN stimuli, including (a) a single spike stimulus, (b) multiple low frequency stimuli, and (c) a tonic current analogous to multiple high frequency stimuli. The chosen stimuli replicate those used for biological data collection in figure 1 of [1].

servo commanded angles and commanded speeds), and then send data sentences to the OpenCM. If a robot command has changed since the last sentence was sent to the robot, then a three-byte packet is added to the next sentence, specifying the ID of the target servo and a two-byte representation of the command. Checking for changes in the commanded values increases the efficiency of communication by reducing how many commands are actually sent to the robot. Converting the commands to the servos' integer-based representation before sending them further reduces the chances of the command changing and needing to be sent. The OpenCM runs an embedded version of Szerial in order to receive and decode messages from AnimatLab, and package and send sensory information back to the neural controller in the same format.

During runtime, both AnimatLab and the OpenCM have a fixed time-step at which to send updates. For the experiments in this manuscript, that duration was 20 ms. If the OpenCM receives a new sentence from AnimatLab, it is parsed into commanded angles and moving speeds for the Dynamixel smart servos, and the commands are broadcasted to the servos using the dynamixelSDK library. If the OpenCM does not receive a new sentence from AnimatLab within the specified period, it proceeds to read sensor values (i.e. servo angles and speeds and strain gauge voltages) and report them back to AnimatLab.

Each strain gauge has its own custom Wheatstone bridge. The OpenCM converts the analog voltage signals into 12-bit digital values. A custom multiplexer enables the OpenCM to read the voltage of 18 separate strain gauges in each control loop. Sensory information from the strain gauges is pre-processed before being transmitted to AnimatLab. Depending on the tuning of the Wheatstone bridge, there will be a constant offset voltage. Since the main purpose of the strain gauges is to determine when a leg is in stance phase, such an offset must be eliminated. To continuously cancel this offset, we implemented the filter described in section 3.2.5 onboard the OpenCM microcontroller. This filter ensures that constant offsets are eliminated over time.

5. Robot similarities to the simulation and animals

5.1. Joint control and response

After the construction of Drosophibot, we performed several tests to ensure that the neural and mechanical systems interface as intended and would produce the biomimetic features listed in the Introduction. First, we verified that Drosophibot's MN activity produces motion in an insect-like way. Specifically,

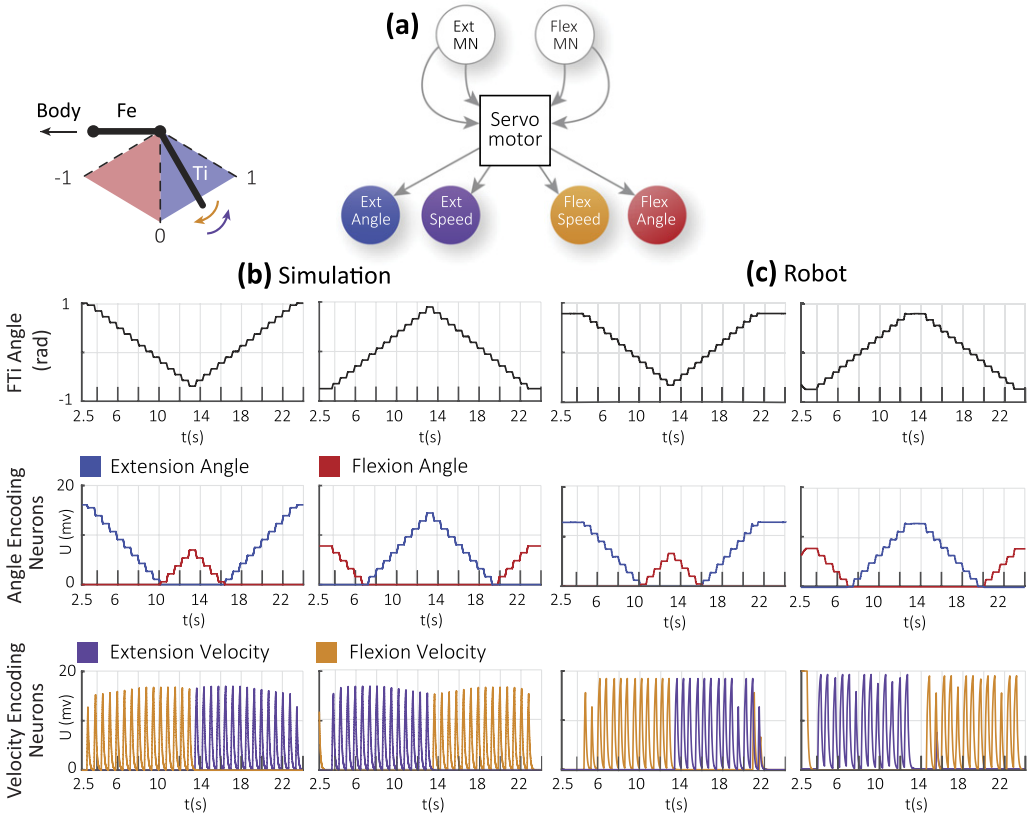


Figure 12. The adapters we have developed between the actuator and the sensory neurons for each joint allow for biomimetic sensory feedback encoding for position and velocity. (a) The layout of the portion of the network which we tested with the neurons of interest color coded. (b) The response of the position and velocity encoding neurons in simulation when given two different incrementally ramping position signal (black). (c) The same responses from the neurons when the network is controlling the robot. In both cases, the input signals were modeled off of those used in biological data collection in [43], which are represented in figure 6.

we expected MN activity to establish the equilibrium angle of the joint [65], and a lack of MN activity to cause the joint to rotate back to an equilibrium angle [1]. To test this, we removed the synapses connecting the CPG of the left front FTi joint to the extension and flexion MNs, isolating the single joint from full network stimulus (figure 4(a)). We then stimulated the joint's flexion MN with repetitive currents of different frequencies and durations, as inspired by [1]. These stimuli were of three types: A short current stimulus (single spike), multiple short current stimuli delivered at a low frequency (slow spiking), and a tonic current (fast spiking). The resulting joint rotation from the various stimuli are shown in figure 11, both in simulation (repeated from figure 5) and on Drosophibot.

In each stimulus case, the robot's limb attempts to passively return to the equilibrium position defined by the elastic elements when the stimulus ends. In the case of a single spike (figure 11(a)), this return results in a quick kick of the limb with a much smaller magnitude than the commanded motion. Multiple pulses actively flex the limb closer to the commanded point, but if the delay between spikes is too great the limb passively extends back to equilibrium in between pulses, resulting in disjointed motion (figure 11(b)). Smooth, sustained limb motion requires dense current pulses or a tonic current (figure 11(c)), during which the joint smoothly flexes, and after which the joint slowly extends back to equilibrium. The robot and simulation responses vary in that the range of motion of the robot is limited to 0.75 radian, and in that the robot's motion lags behind commands by about 100 ms. This delay may be due to real-world physics limiting the rate at which the servomotor can draw current and apply torque to overcome the passive spring torque.

This passive return behavior closely mimics the similar response observed in a variety of insects. Ache and Matheson (2013) in particular used similar MN stimuli on a locust leg joint and reported the same movements in their figure 1 [1]. Thus, the inclusion of parallel elasticity in the robot's joints and manner of converting MN activation into servomotor commands result in similar passive joint behaviors as in insects, supporting Drosophibot's role as a plausible mechanical analog to an insect.

5.2. Proprioception

In addition to tuning the connections from the MNs to the actuator, we also ensured that the adapters between the actuator and the sensory neurons

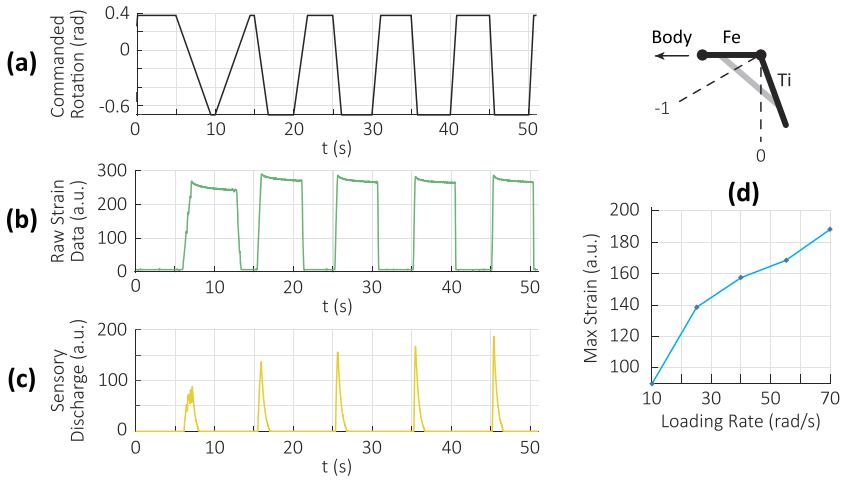


Figure 13. Our method for filtering strain data resembles that found in the sensilla of insects, and allows us to more robustly measure the force on each limb. (a) The oscillating commanded rotation of the FTi joint on Drosophibot, in which the speed of each oscillation incrementally increases (b) the raw strain data associated with the oscillation (c) the results of filtering the strain data with our selected method (d) the relationship between the commanded speed of the attempted motion (referred to as the loading rate) and the corresponding maximum filtered strain value. Note the similarity with the insect data in figure 7.

correctly and biomimetically encode position and velocity feedback. The results of this validation are shown in figure 12. To begin, we isolated single joint control from the rest of the network in the same manner as the test in figure 11. We then stimulated the extensor and flexor MNs with incremental step inputs, causing the joint to incrementally oscillate between rotational extremes. We recorded how the velocity and position sensory neurons encoded these motions in simulation (b) and on the robot (c).

The data from these tests show that the extension position neuron (blue) only activates when the joint is at a positive angle from equilibrium, corresponding to extension of the limb. The level of activation directly correlates to the magnitude of the rotation. The flexion position neuron (red) behaves similarly, only activating when the joint is at a negative angle. Together, these angle encoding neurons record the full motion of the limb. The velocity neurons, meanwhile, activate only while the joint is in motion, and then deactivate once the motion ceases. Which neuron activates corresponds to the direction of motion; the flexion velocity neuron only responds to clockwise motion, and the extension velocity neuron to counterclockwise motion. Such a manner of sensory encoding aligns closely with that of *Drosophila*, as shown in figure 5(a) and (c) of Mamiya *et al* (2018) [43]. With the sensory feedback adapters tuned to encode data biomimetically, we are able to correctly interface the neural and mechanical systems in regards to output and input.

5.3. Strain filtering

In addition to position and velocity data, we also validated the form of feedback from the strain gauges such that the processing adapts to constant loads. Such filtering allows the robot to detect sudden increases in load on the limb, such as at the beginning of stance phase, with more sensitivity. To test this filtering strategy, we recorded the strain of a constrained FTi joint on the robot as we commanded it to oscillate between flexed and relaxed postures at incrementally increasing speeds. Figure 13 shows the raw (b) and filtered (c) strain data as a result of these oscillations (a). The data shows that this method of filtering successfully records the onset of the force on the limb, then begins to decay back to zero as the raw strain data plateaus. As the speed of the motion generating the force increases, the magnitude of the filtered strain spike does as well. Figure 13(d) explicitly shows this relationship. These characteristics of the strain gauge output also closely match the corresponding neural encoding of the campaniform sensilla in the stick insect *Carausius morosus* and the cockroach *Periplaneta americana*, as shown in figure 6 of [83] and figures 5 and 6 of [49], respectively. Ensuring this manner of insect-like force sensing on the robot improves the fidelity of the robot as a model of the insect, as well as the robustness of its sensing capabilities.

6. Discussion

We present neuromechanics models of the fruit fly *D. melanogaster* in the form of a hardware robot and an accompanying simulation. Our goal is to construct a 'morphological' model of insect neuromechnanics [11]: we seek to model what is known about the animal, and then build further hypotheses on top of these features. We intend to use Drosophibot to test dynamical SNS models of neural control [64, 66, 67, 69, 70], as presented in section 3. To ensure that the robot is a meaningful model of the animal's body, we showed that Drosophibot's dynamic scale is consistent with

that of an insect by increasing the ratio of elastic to inertial forces in the body and forcing the robot to move slowly. To ensure that the SNS controller interfaces with the body in a biologically plausible way, we showed that our actuation scheme converts MN activity into joint rotation in the same was as observed in insects, and we showed that our proprioceptive and strain-sensing feedback systems process information as insect sensory afferents do. We showed that these features can be incorporated into a simulation of walking behavior, from which several key features of insect locomotion emerge. Finally, we described the hardware robot Drosophibot and showed that its motor output and sensory feedback function in the same way as the computational model. We believe that Drosophibot's morphological design makes it valuable for testing hypotheses about neural control in hardware.

The morphological approach to bioinspired robotics is challenging because morphological models and robots may sacrifice overall function in the name of biological accuracy [11]. Indeed, abstracting key concepts from biological systems is one way to improve the performance of engineered devices or shed light onto fundamental principles in neuromechanics [24]. However, we believe there is much to be gained by the morphological approach because increased scrutiny of biological details oftentimes yields unexpected solutions. For example, the strain processing system described in section 4.5 not only filters sensory feedback as seen in the insect nervous system; it also solved the persistent problem of calibrating the amplifier's offset such that the sensor is responsive to small forces without a constant bias. By enabling the system to continuously adapt, we made Drosophibot's operation more reliable and simpler for the operator. We believe that more solutions like this one will be discovered as we continue to refine our morphological biorobot.

6.1. Importance of dynamic scaling

If a robot is to serve as a morphological neuromechanical model of an animal, it must be dynamically scaled [32]. Such scaling is necessary to successfully apply an animal's control strategy to a robot. For example, large animals such as humans and horses use a momentum-based strategy to control some motions, wherein they propel their limbs with brief bursts of muscle contraction and rely on inertia to bring the limb to its intended position. In contrast, small animals such as insects need to persistently activate their muscles in order to overcome the elastic forces trying to return their legs to an equilibrium orientation [33]. Clearly, these strategies are not interchangeable. Therefore, biomimetic robots must be constrained to operate in the same dynamic regime as the model organism. However, dynamic scaling also liberates engineers by enabling them to build a robot of any *size*, provided they take steps to match the *dynamic scale*. We ensured that Drosophibot is properly scaled by incorporating additional elastic elements to bias it toward an insect-like, elasticity-dominated regime and by slowing its motions such that they are slower than the natural frequency of its joints. Such an approach enables researchers to build practical models of a diverse range of animals and uncover fundamental similarities and differences between them.

Remarkably, the walking speed of Drosophibot is consistent with the walking speed of the fruit fly, once scaled. At first glance, one might expect the walking speed normalized to body length (BL), $\bar{v}$, to be the same between the robot and the fly. However, the Strouhal number implies that time scales with length, because mass scales with length cubed and stiffness scales with length [38]. Therefore, the distance traveled divided by BL does not scale with length, but the time required for the motion does. Given that a fruit fly can walk at 15 BL per second [61], and Drosophibot is 100 times the length scale of a fruit fly, one would expect Drosophibot to walk at 0.15 BL per second. Given Drosophibot's BL of 35 cm, its walking speed of 5 cm s^{-1} in simulation is precisely what one would expect ($\bar{v} = 5$ cm $s^{-1} \cdot 1$ BL/35 cm $= 0.143$ BL/s). Dynamic scaling is a powerful tool that enables scientists and engineers to compare the dynamics of animals and robots across different time- and length-scales.

6.2. Using Drosophibot to test further networks

The results in this manuscript are meant to show that Drosophibot is a plausible neuromechanical model of an insect, upon which detailed models of the neural control of locomotion can be built. Thus, the results are broad and in some cases preliminary. However, these results will serve as the basis for several future studies.

Our immediate future work is to better understand how the insect's proprioceptive and load-sensing systems process sensory information, and how the nervous system may alter these systems to produce context-dependent sensorimotor control loops. We anticipate that adaptive walking behavior will emerge once we implement more detailed models of sensorimotor processing onboard Drosophibot. Once Drosophibot can produce nominal walking, we will further expand the functionality of the walking control networks. We are particularly interested in how the network can be modified by descending commands to cause locomotion at different speeds and in different directions. Many insect studies show that changes in walking direction [30, 44] and speed [6, 25] arise due to changes in how sensory feedback is processed. Drosophibot's ability to measure joint angles, joint velocities, and strain from three locations on each leg will be critical as we expand this controller to produce more diverse locomotion behaviors.

Furthermore, we wish to understand how brain networks formulate and encode descending commands. The central complex (CX) is an increasingly well-understood brain network whose activity precedes and strongly correlates with locomotion direction [29, 44, 73]. Our recent dynamical model of substructures within the CX demonstrated how it may balance the influence of multiple exteroceptive inputs to 'decide' on one resulting heading [46]. We will continue to expand this model and integrate its outputs with the local leg networks we implemented here to build a plausible model of how the nervous system gathers information from its environment, processes it in a context-dependent way, and directs ongoing behaviors via a distributed control system. Such a model would serve as a substrate on which to model additional animal behaviors and neural pathways as they become better understood.

Another question to address is how flexible interleg coordination rules must be to coordinate walking at different speeds and in different directions. Behavioral studies [13] and computational models [14] suggest that these rules do not need to change, even as an insect or robot changes its locomotion. Indeed, the whole 'continuum of gaits' can be predicted by assuming an interleg coordination scheme that seeks to keep the projection of the COM near the center of the support polygon created by the legs [61, 72]. As we test our computational neuromechanical model in new contexts, we will pay special attention to how the larger context of locomotion may affect what interleg processing is necessary for successful walking.

Despite what can be learned from the simulation, we expect to learn more when it is applied in hardware. For example, the rules that coordinate idealized models may not be necessary or sufficient when tested on board a noisy hardware robot. Recent work in stick insects have shown that mechanical coupling can play a critical role in interleg coordination [16], an effect that we observed in our dynamical simulation of walking as well. However, we may find that this mechanism is enhanced or diminished with a hardware robot. Therefore, we will use Drosophibot to test the sufficiency of established interleg coordination rules, and explore if additional influences are necessary to keep the body upright and moving.

In addition to control, we wish to better understand how the robot's mechanics will affect both its performance and its durability. Drosophibot currently has passively compliant tarsi, which we expect will reduce impact loading on the actuators and help the legs conform to uneven terrain. However, we wish to more thoroughly characterize how tarsus geometry and compliance affect the torques that the actuators must produce. Additionally, insects can actuate their tarsi to actively grip the walking substrate, albeit via scale-dependent mechanisms such as van der Waals forces and capillary action [27]. Can these same mechanisms be exploited for much large (i.e. centimeter-scale) foot pads? Should macroscopic techniques such as spines that embed into a soft substrate be used instead? We plan to use Drosophibot as a testbed with which to test how foot designs affect the performance and lifespan of actuators, in addition to how they affect the control of walking.

Appendix A. Dynamic scaling calculations

To compute $(\mathrm{st})^2$ from equation (1) for the fly, we know that T_{cycle} is 60 ms at its fastest running speed [81]. To approximate k_{joint} for the fly, we can use the calculated value of 38×10^{-6} N m rad^{-1} for the stick insect [33] and use length-based scaling laws [38] to approximate $k_{\mathrm{joint}} = 38 \times 10^{-9}$ N m rad^{-1}. Given that the mass of a fruit fly is about 1.2 mg [81] and that each leg contains about 2% of the body's mass [61], the mass of a leg is about 2.4 μg. Using the formula for the moment of inertia about a fixed point (i.e. the thorax), we obtain an approximate $J_{\mathrm{limb}} = 8 \times 10^{-15}$ for the fly. With these values, we can calculate

$$(\mathrm{st})^2_{\mathrm{fly}} \approx \frac{1}{4\pi^2} \cdot \frac{38 \times 10^{-9}\ \mathrm{Nm\ rad^{-1}}}{8 \times 10^{-15}\ \mathrm{kgm^2}} \cdot (60 \times 10^{-3}\ \mathrm{s})^2 = 43. \quad \text{(A.1)}$$

Such a large value for $(\mathrm{st})^2_{\mathrm{fly}}$ suggests that the fly's locomotion is dominated by elastic forces.

Next, we can use measurements of k_{joint} and J_{limb} to calculate the stepping period at which the robot and the fruit fly have the same dynamic scale. Each joint on the robot possesses a servo that acts like a spring, applying a torque proportional to the deviation from the commanded position. We measured this stiffness to be $k_{\mathrm{joint}} = 34.2$ N m rad^{-1}. Because of the servo's high gear ratio (508:1), the inertia of the rotor dominates the moment of inertia. We measured the moment of inertia of the rotor to be 400×10^{-3} kg m^2. Including the effect of the FTi servo with a mass of 125 g positioned 10 cm away from the thorax increases the moment of inertia to 525×10^{-3} kg m^2 for the whole leg. To ensure that $(\mathrm{st})^2_{\mathrm{fly}} = (\mathrm{st})^2_{\mathrm{robot}}$, $T_{\mathrm{cycle}} = 1.61$ s for the robot. Of course, this value is only approximate, meaning that the robot should step with a cycle time on the order of 1 s. Drosophibot was designed to walk with a cycle of 2 s.

Appendix B. Neural and synaptic model

The entirety of Drosophibot's control system is constructed from nonspiking leaky compartments. The model we use and associated design tools are detailed

elsewhere [66, 67]. However, a brief summary will be provided here.

Most neurons in the network have only one dynamical variable U, the voltage above the rest voltage. Assuming a membrane conductance of 1, U evolves according to

$$C_{\text{mem}} \cdot \frac{\mathrm{d}U}{\mathrm{d}t} = -U + \sum_{i=1}^{n} G_{\text{syn},i} \cdot (\Delta E_{\text{syn},i} - U) + G_{\text{NaP}} \cdot m \cdot h \cdot (\Delta E_{\text{NaP}} - U) + I_{\text{app}}. \tag{B.1}$$

All G terms are conductances, all ΔE terms are the reversal potential of a synapse or ion channel relative to the neuron's rest potential, and C_{mem} is the capacitance of the cell membrane. The subscript *syn,i* refers to the ith incoming synapse, the subscript *NaP* refers to a persistent sodium channel, I_{app} is a constant applied current (usually 0). If $G_{\text{NaP}} \neq 0$, then the channel activation m and channel deactivation h evolve according to

$$\tau_m(U) \cdot \frac{\mathrm{d}m}{\mathrm{d}t} = m_\infty(U) - m \tag{B.2}$$

$$\tau_h(U) \cdot \frac{\mathrm{d}h}{\mathrm{d}t} = h_\infty(U) - h, \tag{B.3}$$

where $m_\infty(U)$ and $h_\infty(U)$ are sigmoids with positive and negative slope, respectively. In addition, $\tau_m \ll \tau_h$. Therefore, m and h deliver rapid positive feedback followed by slow negative feedback.

Synapses are conductance based, as shown in equation (B.1). The conductance is a piecewise-linear function of the presynaptic neuron's voltage above rest,

$$G_{\text{syn},i} = g_{\text{max},i} \cdot \begin{cases} 1 & \text{if } U_{\text{pre},i} > R \\ \dfrac{1}{R} & \text{if } 0 \leqslant U_{\text{pre},i} \leqslant R \\ 0 & \text{if } U_{\text{pre},i} < 0 \end{cases} \tag{B.4}$$

where $g_{\text{max},i}$ is the maximum conductance of the ith synapse, and R is a parameter that represents the expected voltage range of the neurons in a particular network. Establishing the parameter R aids in the direct assembly and tuning of small functional networks whose encoded quantities are clear, and can therefore be directly interfaced with one another [66].

ORCID iDs

C A Goldsmith https://orcid.org/0000-0002-3193-520X
N S Szczecinski https://orcid.org/0000-0002-6453-6475

References

[1] Ache J M and Matheson T 2013 Passive joint forces are tuned to limb use in insects and drive movements without motor activity *Curr. Biol.* **23** 1418–26
[2] Aschenbeck K S, Kern N I, Bachmann R J and Quinn R D 2006 Design of a quadruped robot driven by air muscles *Proceedings of the First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006, BioRob 2006* 875–80
[3] Ayali A, Borgmann A, Büschges A, Couzin-Fuchs E, Daun-Gruhn S and Holmes P 2015 The comparative investigation of the stick insect and cockroach models in the study of insect locomotion *Current Opinion in Insect Science* **12** 1–10
[4] Bässler U 1988 Functional principles of pattern generation for walking movements of stick insect forelegs: the role of the femoral chordotonal organ afferences *J. Exp. Biol.* **136** 125–47
[5] Belter D and Walas K 2014 *A Compact Walking Robot—Flexible Research and Development Platform BT—Recent Advances in Automation, Robotics and Measuring Techniques* (Berlin: Springer) pp 343–52
[6] Berendes V, Zill S N, Büschges A and Bockemühl T 2016 Speed-dependent interplay between local pattern-generating activity and sensory signals during walking in Drosophila *J. Exp. Biol.* **219** 3781–93
[7] Bledt G, Powell M J, Katz B, Di Carlo J, Wensing P M and Kim S 2018 MIT Cheetah 3: design and control of a robust, dynamic quadruped robot *IEEE International Conference on Intelligent Robots and Systems (December)* pp 2245–52
[8] Blickhan R and Full R J 1993 Similarity in multilegged locomotion: Bouncing like a monopode *J. Comp. Physiol.*, A **173** 509–17
[9] Büschges A, Ludwar B C, Bucher D, Schmidt J and a DiCaprio R 2004 Synaptic drive contributing to rhythmic activation of motoneurons in the deafferented stick insect walking system *The European Journal of Neuroscience* **19** 1856–62
[10] Büschges A, Schmitz J and Bässler U 1995 Rhythmic patterns in the thoracic nerve cord of the stick insect induced by pilocarpine *J. Exp. Biol.* **198** 435–56
[11] Buschmann T, Ewald A, Von Twickel A and Büschges A 2015 Controlling legs for locomotion - insights from robotics and neurobiology *Bioinspiration Biomimetics* **10** 41001
[12] Cofer D W, Cymbalyuk G, Reid J, Zhu Y, Heitler W J and Edwards D H 2010 AnimatLab: a 3D graphics environment for neuromechanical simulations *J. Neurosci. Methods* **187** 280–8
[13] Cruse H 1990 What mechanisms coordinate leg movement in walking arthropods? *Trends Neurosci.* **13** 15–21
[14] Cruse H, Kindermann T, Schumm M, Dean J and Schmitz J 1998 Walknet-a biologically inspired network to control six-legged walking *Neural networks : the official journal of the International Neural Network Society* **11** 1435–47
[15] Dallmann C J, Dürr V and Schmitz J 2016 Joint torques in a freely walking insect reveal distinct functions of leg joints in propulsion and posture control *Proc. R. Soc.* B **283** 20151708
[16] Dallmann C J, Hoinville T, Du V and Schmitz J 2017 A load-based mechanism for inter-leg coordination in insects *Proc. R. Soc.* B **264** 20171755
[17] Dasgupta S, Goldschmidt D, Worgotter F and Manoonpong P 2015 Distributed recurrent neural forward models with synaptic adaptation and CPG-based control for complex behaviors of walking robots *Frontiers in Neurorobotics* **9** 10
[18] Delcomyn F 1971 Computer aided analysis of a locomotor leg reflex in the cockroach *Periplaneta americana Z. Vgl. Physiol.* **74** 427–45

[19] Dirk S and Frank K 2007 The bio-inspired SCORPION robot: design, control & lessons learned *Climbing and Walking Robots: Towards New Applications* (London: IntechOpen Limited) pp 197–218

[20] Dürr V *et al* 2019 Integrative biomimetics of autonomous hexapedal locomotion *Frontiers in Neurorobotics* **13** 1–32

[21] Eckert P, Schmerbauch A E M, Horvat T, Söhnel K, Fischer M S, Witte H and Ijspeert A J 2019 Towards rich motion skills with the lightweight quadruped robot Serval *Adapt. Behav.* **28** 129–50

[22] Espenschied K S, Quinn R D, Beer R D and Chiel H J 1996 Biologically based distributed control and local reflexes improve rough terrain locomotion in a hexapod robot *Robot. Auton. Syst.* **18** 59–64

[23] Fukuoka Y and Kimura H 2009 Dynamic locomotion of a biomorphic quadruped 'Tekken' robot using various gaits: Walk, trot, free-gait and bound *Applied Bionics and Biomechanics* **6** 63–71

[24] Full R J and Koditschek D E 1999 Templates and anchors: neuromechanical hypotheses of legged locomotion on land *J. Exp. Biol.* **202** 3325–32

[25] Gabriel J P and Büschges A 2007 Control of stepping velocity in a single insect leg during walking *Phil. Trans. R. Soc.* **365** 251–71

[26] Goldschmidt D, Wörgötter F and Manoonpong P 2014 Biologically-inspired adaptive obstacle negotiation behavior of hexapod robots *Frontiers in Neurorobotics* **8** 1–3

[27] Gorb S N 2005 . Uncovering insect stickiness: structure and properties of hairy attachment devices *American Entomologist* **51** 31–5

[28] Görner M, Wimböck T, Baumann A, Fuchs M, Bahls T, Grebenstein M, Borst C, Butterfass J and Hirzinger G 2008 The DLR-crawler: a testbed for actively compliant hexapod walking based on the fingers of DLR-hand II *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS* pp 1525–31

[29] Heinze S 2017 Neural coding: bumps on the move *Curr. Biol.* **27** R409–12

[30] Hellekes K, Blincow E, Hoffmann J and Büschges A 2011 Control of reflex reversal in stick insect walking: effects of intersegmental signals, changes in direction and optomotor induced turning *J. Neurophysiol.* **107** 239–49

[31] Hess D and Büschges A 1997 Sensorimotor pathways involved in interjoint reflex action of an insect leg *J. Neurobiol.* **33** 891–913

[32] Hooper S L 2012 Body size and the neural control of movement *Curr. Biol.* **22** R318–22

[33] Hooper S L, Guschlbauer C, Blümel M, Rosenbaum P, Gruhn M, Akay T and Büschges A 2009 Neural control of unloaded leg posture and of leg swing in stick insect, cockroach, and mouse differs from that in larger animals *J. Neurosci.* **29** 4109–19

[34] Hutter M *et al* 2016 ANYmal—a highly mobile and dynamic quadrupedal robot *IEEE International Conference on Intelligent Robots and Systems* (2016-November 38–44)

[35] Ijspeert A J 2014 Biorobotics: using robots to emulate and investigate agile locomotion *Science* **346** 196–203

[36] Jan Ijspeert A 2008 Central pattern generators for locomotion control in animals and robots: a review *Neural Netw.* **21** 642–53

[37] Ijspeert A J, Crespi A, Ryczko D and Cabelguen J-M 2007 From swimming to walking with a salamander robot driven by a spinal cord model *Science* **315** 1416–20

[38] Jayaram K, Mongeau J M, Mohapatra A, Birkmeyer P, Fearing R S and Full R J 2018 Transition by head-on collision: mechanically mediated manoeuvres in cockroaches and small robots *J. R. Soc., Interface* **15** 20170664

[39] Karakasiliotis K, Thandiackal R, Melo K, Horvat T, Mahabadi N K, Tsitkov S, Cabelguen J M and Ijspeert A J 2016 From cineradiography to biorobots: an approach for designing robots to emulate and study animal locomotion *J. R. Soc., Interface* **13** 20151089

[40] Kingsley D A, Quinn R D and Ritzmann R E 2006 A cockroach inspired robot with artificial muscles *IEEE International Conference on Intelligent Robots and Systems* pp 1837–42

[41] Lewinger W A and Quinn R D 2010 A hexapod walks over irregular terrain using a controller adapted from an insect's nervous system *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010—Conference Proceedings* pp 3386–91

[42] Lewinger W A, Martin Reekie H and Webb B 2011 A hexapod robot modeled on the stick insect, Carausius morosus *IEEE 15th International Conference on Advanced Robotics: New Boundaries for Robotics, ICAR 2011* pp 541–8

[43] Mamiya A, Gurung P and Tuthill J C 2018 Neural coding of leg proprioception in Drosophila *Neuron* **100** 636–650.e6

[44] Martin J P, Guo P, Mu L, Harley C M and Ritzmann R E 2015 Central-complex control of movement in the freely walking cockroach *Curr. Biol.* **25** 2795–803

[45] Mendes C S, Bartos I, Akay T, Marka S and Mann R S 2013 Quantification of gait parameters in freely walking wild type and sensory deprived *Drosophila melanogaster eLife* **2** e00231

[46] Pickard S C, Quinn R D and Szczecinski N S 2019 A dynamical model exploring sensory integration in the insect central complex substructures *Bioinsp. Biomim.* **15** 0–22

[47] Pratt G A and Williamson M M 1995 Series elastic actuators *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots* vol 1 pp 399–406

[48] Raibert M H 1990 Trotting, pacing and bounding by a quadruped robot *J. Biomech.* **23** 79–81, 83–98

[49] Ridgel A L, Frazier S F, DiCaprio R A and Zill S N 2000 Encoding of forces by cockroach tibial campaniform sensilla: implications in dynamic control of posture and locomotion *J. Comp. Physiol.,* A **186** 359–74

[50] Ritzmann R E, Quinn R D and Fischer M S 2004 Convergent evolution and locomotion through complex terrain by insects, vertebrates and robots *Arthropod Struct. Dev.* **33** 361–79

[51] Ritzmann R E, Quinn R D and Watson J T 2000 Insect walking and biorobotics: a relationship with mutual benefits *Bioscience* **50** 23–33

[52] Robinson D W, Pratt J E, Paluska D J and Pratt G A 1999 Series elastic actuator development for a biomimetic walking robot *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.99TH8399)* pp 561–8

[53] Roennau A, Heppner G, Nowicki M and Dillmann R 2014 LAURON V: a versatile six-legged walking robot with advanced maneuverability *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* pp 82–7

[54] Rosendo A, Nakatsu S, Narioka K and Hosoda K 2013 Pneupard: a biomimetic musculoskeletal approach for a feline-inspired quadruped robot *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* pp 1452–7

[55] Rubeo S, Szczecinski N and Quinn R 2017 A synthetic nervous system controls a simulated cockroach *Appl. Sci.* **8** 6

[56] Rutishauser S, Spröwitz A, Righetti L and Ijspeert A J 2008 Passive compliant quadruped robot using central pattern generators for locomotion control *Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob 2008* pp 710–5

[57] Sauer A E, Driesang R B, Bueschges A and Baessler U 1995 Information processing in the femur-tibia control loop of stick insects *J. Comp. Physiol.,* A **177** 145–58

[58] Saunders F, Trimmer B A and Rife J 2011 Modeling locomotion of a soft-bodied arthropod using inverse dynamics *Bioinsp. Biomim.* **6** 016001

[59] Scharzenberger C, Mendoza J and Hunt A 2019 *Design of a Canine Inspired Quadruped Robot as a Platform for Synthetic*

Neural Network Control BT—Biomimetic and Biohybrid Systems (Berlin: Springer) pp 228–39
[60] Spenko M J, Haynes G C, Aaron Saunders J, Cutkosky M R, Rizzi A A, Full R J and Koditschek D E 2008 Biologically inspired climbing with a hexapedal robot *Journal of Field Robotics* **25** 223–42
[61] Szczecinski N S, Bockemühl T, Chockley A S and Büschges A 2018 Static stability predicts the continuum of interleg coordination patterns in *Drosophila J. Exp. Biol.* **221** jeb189142
[62] Szczecinski N S, Brown A E, Bender J A, Quinn R D and Ritzmann R E 2014 A neuromechanical simulation of insect walking and transition to turning of the cockroach *Blaberus discoidalis Biol. Cybern.* **108** 1–21
[63] Szczecinski N S, Chrzanowski D M, Cofer D W, Terrasi A S, Moore D R, Martin J P, Ritzmann R E and Quinn R D 2015 Introducing MantisBot: Hexapod robot controlled by a high-fidelity, real-time neural simulation *IEEE International Conference on Intelligent Robots and Systems* (2015-December) pp 3875–81
[64] Szczecinski N S, Getsy A P, Martin J P, Ritzmann R E and Quinn R D 2017 MantisBot is a robotic model of visually guided motion in the praying mantis *Arthropod Struct. Dev.* **46** 736–51
[65] Szczecinski N S, Goldsmith C A, Young F R and Quinn R D 2019 Tuning a robot servomotor to exhibit muscle-like dynamics *Biomimetic and Biohybrid Systems* ed U Martinez-Hernandez, V Vouloutsi, A Mura, M Mangan, M Asada, T J Prescott and P F M J Verschure (Berlin: Springer) pp 254–65
[66] Szczecinski N S, Hunt A J and Quinn R D 2017 A functional subnetwork approach to designing synthetic nervous systems that control legged robot locomotion *Frontiers in Neurorobotics* **11** 37
[67] Szczecinski N S, Hunt A J and Quinn R D 2017 Design process and tools for dynamic neuromechanical models and robot controllers *Biol. Cybern.* **111** 105–27
[68] Szczecinski N S and Quinn R D 2017 Leg-local neural mechanisms for searching and learning enhance robotic locomotion *Biol. Cybern.* **112** 99–112
[69] Szczecinski N S and Quinn R D 2017 MantisBot changes stepping speed by entraining CPGs to positive velocity feedback *Lecture Notes in Artificial Intelligence 10384* pp 440–52
[70] Szczecinski N S and Quinn R D 2017 Template for the neural control of directed stepping generalized to all legs of MantisBot *Bioinsp. Biomim.* **12** 045001
[71] Theunissen L M, Bekemeier H H and Dürr V 2015 Comparative whole-body kinematics of closely related insect species with different body morphology *J. Exp. Biol.* **218** 340–52
[72] Ting L H, Blickhan R and Full R J 1994 Dynamic and static stability in hexapedal runners *J. Exp. Biol.* **197** 251–69
[73] Varga A G, Kathman N D, Martin J P, Guo P and Ritzmann R E 2017 Spatial Navigation and the Central Complex: Sensory Acquisition, Orientation, and Motor Control *Front. Behav. Neurosci.* **11** 1–4
[74] Von Twickel A, Hild M, Siedel T, Patel V and Pasemann F 2012 Neural control of a modular multi-legged walking machine: Simulation and hardware *Robot. Auton. Syst.* **60** 227–41
[75] Watson J T and Ritzmann R E 1998 Leg kinematics and muscle activity during treadmill running in the cockroach, *Blaberus discoidalis*: I. Slow running *J. Comp. Physiol.,* A **182** 11–22
[76] Watson J T and Ritzmann R E 1998 Leg kinematics and muscle activity during treadmill running in the cockroach, *Blaberus discoidalis*: II. Fast running *J. Comp. Physiol.,* A **182** 23–33
[77] Watson J T, Ritzmann R E and Pollack A J 2002 Control of climbing behavior in the cockroach, *Blaberus discoidalis*. II. Motor activities associated with joint movement *J. Comp. Physiol.,* A **188** 55–69
[78] Wilson H R and Cowan J D 1972 Excitatory and inhibitory interactions in localized populations of model neurons *Biophys. J.* **12** 1–24
[79] Wolf H and Büschges A 1995 Nonspiking local interneurons in insect leg motor control. II. Role of nonspiking local interneurons in the control of leg swing during walking *J. Neurophysiol.* **73** 1861–75
[80] Wolpert D M and Ghahramani Z 2000 Computational principles of movement neuroscience *Nat. Neurosci.* **3** 1212–7
[81] Wosnitza A, Bockemuhl T, Dubbert M, Scholz H and Buschges A 2013 Inter-leg coordination in the control of walking speed in *Drosophila J. Exp. Biol.* **216** 480–91
[82] Zakotnik J, Matheson T and Dürr V 2006 Co-contraction and passive forces facilitate load compensation of aimed limb movements *The Journal of Neuroscience* **26** 4995LP–495007
[83] Zill S N, Büschges A and Schmitz J 2011 Encoding of force increases and decreases by tibial campaniform sensilla in the stick insect, *Carausius morosus J. Comp. Physiol.,* A **197** 851–67
[84] Zill S N, Dallmann C J, Büschges A, Chaudhry S and Schmitz J 2018 Force dynamics and synergist muscle activation in stick insects: the effects of using joint torques as mechanical stimuli *J. Neurophysiol.* **120** 1807–23
[85] Zill S N, Schmitz J and Büschges A 2004 Load sensing and control of posture and locomotion *Arthropod Struct. Dev.* **33** 273–86
[86] Zill S N, Schmitz J, Chaudhry S and Büschges A 2012 Force encoding in stick insect legs delineates a reference frame for motor control *J. Neurophysiol.* **108** 1453–72