Bioinspiration &
Biomimetics

PURPOSE-LED
PUBLISHING™

PAPER • OPEN ACCESS

Walking on rough terrain with any number of legs

To cite this article: Zhuoyang Chen *et al* 2026 *Bioinspir. Biomim.* **21** 046040

View the article online for updates and enhancements.

You may also like

- Cascading reconfigurable skyrmion logic devices: simplified architectures for versatile computing systems
Sarwath Sara, Chandrasekhar Murapaka and Arabinda Haldar
- Combustion-derived air pollution and heat from gas-fired appliances in commercial kitchens can exceed health-based guidelines
Yannai S Kashtan, Eric D Lebel, Nicole Lucha et al.
- Quasi-static compression of FDM-printed rotating-square auxetic cores: relative density dominates over rotation angle
Volkan Arkan, Türker Türkolu and Ahmet Çar Klnç

This content was downloaded from IP address 141.211.4.224 on 14/11/2026 at 04:24

Bioinspiration & Biomimetics

Check for updates

OPEN ACCESS

RECEIVED
27 December 2025

REVISED
21 June 2026

ACCEPTED FOR PUBLICATION
29 July 2026

PUBLISHED
26 August 2026

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

Walking on rough terrain with any number of legs

Zhuoyang Chen[1], **Xinyuan Wang**[2] and **Shai Revzen**[1,3,*]

[1] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, United States of America
[2] Department of Robotics, University of Michigan, Ann Arbor, MI, United States of America
[3] Department of Ecology and Evolutionary Biology, University of Michigan, Ann Arbor, MI, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: shrevzen@umich.edu, janchen@umich.edu and wxinyuan@umich.edu

Keywords: finite state machine (FSM), bio-inspired robot, central pattern generator (CPG), legged robot, multi-legged locomotion

Supplementary material for this article is available online

Abstract

Robotics would gain by replicating the remarkable agility of arthropods in navigating complex environments. In this simulation study, we consider the control of 'multi-legged' systems which have 6 or more legs. Current multi-legged control strategies in robots include large black-box machine learning models, central pattern generator (CPG) networks, and open-loop feed-forward control with stability arising from the mechanics. Here we present a multi-legged control architecture for rough terrain using a segmental robot with 3 actuators for every 2 legs, which we validated in simulation for 6 to 16 legs. Segments have identical state machines, and each segment also receives input from the segment in front of it. Our design bridges the gap between Walknet-like event cascade controllers and CPG-based controllers: it tightly couples to the ground when present, but produces fictive locomotion when ground contact is missing. It may be useful as an adaptive, computationally light-weight controller for multi-legged robots, and as baseline capability for scaffolding the learning of machine learning controllers.

1. Introduction

Animals exhibit remarkable abilities to move efficiently through complex and unpredictable environments. Cockroaches and stick insects are classic examples of robust hexapod locomotion on rough terrain (Holmes *et al* 2006). These observations have inspired decades of research in biomechanics, neurophysiology, and bio-inspired robotics.

1.1. Biological inspiration:central pattern generators (CPGs) and Walknet

Early physiological studies on CPGs established that vertebrates and many invertebrates possess neural circuits capable of producing rhythmic activity without requiring step-by-step sensory input (Wilson 1961, Pearson 1972, Edgerton *et al* 1976, Marder and Bucher 2001, Hultborn and Nielsen 2006, Bidaye *et al* 2018), forming the core mechanism underlying walking, swimming, and flying.

Complementary to CPG-based explanations, the Walknet model provides a biologically grounded, decentralized account of six-legged locomotion. Walknet shows that stable insect gaits can emerge without strong central oscillators, relying instead on distributed sensory-driven coordination (Cruse *et al* 1998, Schilling *et al* 2013). Through local rules and the exploitation of body-environment coupling, such as positive velocity feedback, Walknet naturally reproduces tripod and tetrapod gait patterns, adapts to uneven terrain, and remains robust to disturbances such as leg overload, loss of ground contact, or even leg removal. In this sense, Walknet supplements CPG theories by illustrating that rhythmic stability can arise purely from sensory-driven, distributed control in closed loop with the environment (Schilling *et al* 2013). At the same time, the model is compatible with the possibility that central oscillators may take over in conditions where sensory input is compromised, allowing for the possibility that insects flexibly combine decentralized reflexive control and

© 2026 The Author(s). Published by IOP Publishing Ltd

central pattern generation depending on environmental demands (Neveln *et al* 2019).

1.2. Hexapod locomotion: RHex and related platforms

For multi-legged robots, stability can arise mechanically, allowing for feed-forward control strategies and robustness in the face of diverse and unpredictable environments. Some of the earliest work on hexpedal robots hearkens back to McGhee's 'OSU Hexapod[4]' and his work on the late 1960-s on state-machine gait controllers McGhee (1967). At the time, there was great interest in these ideas from the other side of the Iron Curtain as well Okhotsimski and Platonov (1973). Interest in hexapedal robots continued over the decades, with many studies exploiting their natural stability and exploring insect-inspired ideas. Early examples such as **Robot III** and **Biobot** (Nelson *et al* 1997) directly modeled the cockroach leg. The **LAURON I–V** series drew inspiration from the stick insect (Roennau *et al* 2013). In the mid 2000-s **RiSE** (Autumn *et al* 2005, Haynes and Rizzi 2006) wove together ideas from gecko attachment to surfaces, the use of tails for pitch control, and the advantages of 6 legs to produce an exceptional climbing robot that could also walk. Among the many bio-inspired hexapods, **RHex** stands out as one of the most compelling and often replicated examples (Altendorfer *et al* 2001, Saranli *et al* 2001). Renowned for its speed and robustness, RHex can traverse fractured, obstacle-dense, and irregular terrain without sensing or adaptation. Additional work further explored RHex's dynamics, such as self-righting (Saranli *et al* 2004), stair descent (Campbell and Buehler 2003), and later variants included features such as transformable legs (Lin *et al* 2021). Despite its success, RHex also exhibits clear limitations: having 1-DoF per leg leaves it with little freedom for selecting footholds or adapting its motion to the shape of the terrain (Altendorfer *et al* 2001, Campbell and Buehler 2003). The lack of rich sensory feedback further limits its performance in perception-dependent tasks. Beyond hexapods, bio-inspired designs have also explored larger leg counts: for example, Zhang *et al* (2022) present a crab-inspired eight-legged robot whose compliant leg morphology supports adaptive locomotion across varied terrains.

1.3. CPG-based control methods and learning approaches

In motor control research, bio-inspired '*CPGs*' have been broadly used across robots that swim, walk, and slither (Ijspeert and Crespi 2007, Crespi and Ijspeert 2008, Rutishauser *et al* 2008). Comprehensive reviews such as (Ijspeert 2008) and representative works (Klaassen *et al* 2002, Klaassen2002, AInagaki2003, rena2004), (Klaassen2002, AInagaki2003, rena2004) demonstrate the effectiveness of CPGs in generating stable rhythmic locomotion. CPG, used as a control method, are typically modeled as a system of coupled oscillators, with each joint assigned an oscillator. Through appropriate coupling, the system produces stable limit-cycle behaviors that can rapidly recover from perturbations, enabling robust gait generation. Researchers have demonstrated open-loop CPG systems achieving fast locomotion even on rough terrain (Rutishauser *et al* 2008, Sprowitz *et al* 2011). Later studies incorporated sensory feedback to realize closed-loop CPG control (Righetti and Ijspeert 2008). Sartoretti *et al* proposed an approach that adapts CPG parameters using inertial sensory feedback (Sartoretti *et al* 2018). By leveraging body pose, orientation, and height information, the limit cycles of coupled oscillators can adjust themselves in joint space to enable locomotion and body posture control. Various oscillator models—including Hodgkin–Huxley, Kuramoto, Hopf and van der Pol oscillators (Inagaki *et al* 2003, Arena *et al* 2004, Veskos and Demiris 2005, Righetti and Ijspeert 2008)—have been implemented for CPG control. While effective, CPG-based approaches generally lack systematic design principles: coupling topology, phase biases, and gains must often be tuned manually or optimized via heuristics (Ijspeert 2008).

Recently some researchers have proposed learning-based CPG optimization methods (Bellegarda and Ijspeert 2022, Zhang *et al* 2024), and even methods blending CPG structures with modern deep reinforcement learning (Liu *et al* 2020, Deshpande *et al* 2023). The lead to the breakthrough result of a fully learned locomotion controller demonstrated on the **ANYmal** robot (Lee *et al* 2020), highlighting the potential of learning-driven locomotion, albeit at the cost of heavy computation and large-scale data requirements.

1.4. State-machine control

While complex, compute- and sensing-intensive control approaches have produced amazing results, other than the work on RHex, we know little about what simple control and low complexity can accomplish; the pioneering work of the late 1960-s and early 1970-s McGhee (1967), Okhotsimski and Platonov (1973)has lain fallow.

Considering how easily CPG-based control dynamics adapt to alternative oscillator types, we reconsidered whether oscillators with continuous states are necessary altogether. Instead of a continuous state CPG, we discretized the essential functionality of a CPG into a cycle of four task-relevant states and implemented them through a finite state machine

[4] https://cyberneticzoo.com/walking-machines/1976-osu-hexapod-mcghee-american/.

(FSM). This formulation preserved features of rhythmic control, such as phase progression and modularity, while avoiding the complexity of oscillator dynamics, the need for extensive parameter tuning, and the related requirement of high training costs.

We then coupled a segmental state-machine oscillator producing yaw to two leg oscillators raising and lowering the legs. Finally, we allowed each segmental oscillator to influence the oscillator directly behind it. The resulting controller remained simple in structure, but it is capable of generating coordinated and adaptive swing–stance behaviors on irregular and unpredictable terrain. On the mechanical side, we addressed some of the limitations of RHex's leg design with a minimalistic extension. To strike a balance between 'structural simplicity' and 'sufficient locomotive capability,' we introduced a design that has 3 actuators for every pair of legs. This allowed each foot to move within a hemispherical region, which enabled foot placement to be adapted to the terrain features. Finally, we added a Walknet inspired sensory ability—the ability to detect whether a leg has contacted the ground and is bearing load.

2. Methods

While we have built physical versions of our robot(s), the present publication focuses on simulation analysis of the control architecture, and what we may learn from it. We thus use the term 'robot' to refer interchangeably to both physical and simulated robots, and the term 'experiment' to both physical experiments and simulation runs.

2.1. Hardware design

In this paper, we describe actuator layouts using an 'axis-labeling' notation where letters denote the axis of rotation of an actuator relative to the robot body frame at a reference configuration extended along the X axis, and the order of the letters indicates the order of these axes along the body. In our design, the Z axis corresponds to the vertical yaw axis of the robot 'backbone', while the X axis corresponds to the roll axis used to lift or lower a leg. Thus, a configuration written as ZXZ indicates a sequence of actuators consisting of a yaw joint, followed by a roll joint, followed by another yaw joint. Denoting by $(\cdots)$ actuators parallel in the kinematic chain, a $Z(XX)$ configuration describes a single yaw actuator followed by two roll actuators that are parallel to each other, e.g for controlling the right and left legs. Finally, we use $[\cdots]^N$ to indicate a repetition of the bracketed structure N times.

We designed the hexapedal version of our robots (figure 1) as an 8 actuator system inspired by prior 'centipede'-based designs (Sastra *et al* 2008, 2012). These prior robots relied on two spine actuators and one leg actuator in a sequence of repeating ZXZ segments, one for every pair of contra-lateral legs. Because initial and final Z axes have nothing to act on, they can be elided, leaving $XZZXZZX$ as the structure of the hexapedal robot and, more generally, $[XZZ]^N X$ as a family of 'multipod' robot designs with $N+1$ segments, $2N+2$ legs, and $3N+1$ actuators (Zhao 2021). The advantage of this design was that it allowed for non-slip locomotion since its planar projection was a 4-bar linkage between each pair of stance legs (see (Zhao and Revzen 2020, Zhao 2021) for detailed analysis).

However, in the multipod design family one actuator moved both legs on the same segment, which prevented independent height control of legs. Instead of the ZXZ segments of the multipods, we used a $Z(XX)$ configuration: a single backbone yaw motor, followed by a two roll-axis motors in parallel, one to the right and one to the left. This configuration decoupled bilateral leg actuation allowing us to independently control the heights of the right and left foot, at the expense of making slipping inevitable. The family of robot designs we explore here is $(XX)\,[Z(XX)]^N$, with the similar metrics: $N+1$ segments, $2N+2$ legs, but $3N+2$ actuators. The hexapedal version is $(XX)Z(XX)Z(XX)$.

Following the bio-inspired approach in (Sastra *et al* 2008, 2012), we employed compliant legs whose stiffness we chose to match the stiffness appearing in the Spring-Loaded Inverted Pendulum (SLIP) template (Cavagna *et al* 1977, Blickhan and Full 1993) of running animals (Dickinson *et al* 2000). We chose the cantilever properties such that (1) standing on 3 legs, each at roll angle 45°, the robot would sink 10% of its unloaded height; and (2) in this position, the aspect ratio length to width was approximately that of cockroaches.

We designed all structural components with simple geometries suitable for laser cutting and engraving, eschewing slow and fracture prone 3D printing alternatives. We assembled the body segments primarily through geometric mating constraints rather than fasteners; we used screws only for motor mounting and compliant-joint connections, and ensured that all the screws were of the same family (M3 socket head). By chaining additional backbone joints, we could extend the platform from a hexapod to an octopod or even longer configurations (figure 1).

The physical robot we built (figure 1 rightmost) had the segments rigidly linked to each other.

However, to allow for longer robots on rough terrain in the simulation, we included backbone bending joint which is somewhat compliant on the Y axis (pitch), but rigid in Z (yaw) and X (twist along body axis). These backbone bending elements provided an elastic, passive degree of freedom in the

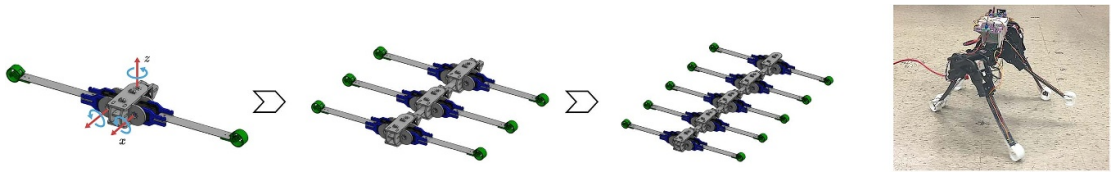


Figure 1. Modular mechanical design of the robot. The segment (left) is $Z(XX)$ and has 3 actuators: two leg motors rotate about the X-axis, followed by one yaw motor that rotates about the Z-axis. This modular design can be chained to create a hexapod $(XX)Z(XX)Z(XX)$ (middle), or, e.g. a decapod $(XX)[Z(XX)]^4$(right). We built a hexapedal version for future physical experiments (right, photo).

pitch direction while avoiding a freedom in roll which would have neutralized much of ability of the legs to lift off. Indicating by lowercase y this passive joint, the simulated robot family we discuss has the structure $(XX)[Zy(XX)]^N$.

Such a backbone element is challenging to build in practice because of the unusually high torsional stiffness required, and its design and fabrication will be the topic of a future publication.

We performed all CAD work using Autodesk Inventor and Onshape; the CAD files are publicly available[5].

2.2. Locomotion control

Our controllers were FSMs designed to operate as closed-loop systems by integrating actuator angles and foot contact sensing into their state transitions. In each segment, they comprised a segmental 'yaw oscillator' with 4 states, including two composite states containing 3 sequential sub-states, respectively, controlling the yaw motor, and 'right leg' and 'left leg' 4-state oscillators, each controlling its respective leg. Figure 2 illustrates the structure of the controller state-machine, including all transitions. The red states represent the yaw-joint FSM, while the gray states represent the left and right leg FSMs. We denoted the leg roll angles as ϕ and the yaw angle as ψ. We defined positive leg swing as upward, and positive yaw rotation as clockwise when viewed from above.

We coupled the yaw controllers unidirectionally from head to tail through a `SYNC` state that blocked execution until the preceding segment entered the composite `STROKE` state, which is divided into three sequential sub-states {`WAIT RISE`, `SWING`, `WAIT FALL`}. This encouraged a stable phase offset between consecutive segments thereby creating a coordinated traveling wave along the body. Each yaw FSM also hierarchically modulated its two leg FSMs. When the yaw FSM entered the `STROKE` state, it activated the leg FSMs to initiate lift-off. The yaw FSM temporarily remained in `WAIT RISE` sub-state to allow each leg FSM to reach the `READY` state once its roll angle ϕ exceeded a threshold. If the yaw FSM exited `STROKE` prematurely, the leg FSM transitioned immediately to `FALL`. During the transition from sub-state `SWING` to sub-state `WAIT FALL`, the yaw FSM triggered the leg to complete foot touchdown. Upon contact detection or ϕ falling below the lower limit, the leg returned to `STAND` and updated its desired position with a small offset Δ_ϕ to produce a support force.

2.2.1. Liveliness in Yaw

Below we prove that under reasonable assumptions, our yaw FSM controller is 'lively' in the sense that the yaw state machine has terrain independent dwell times. Because of the limited transitions available to each of its states, it follows that the 4 yaw oscillator states `RT STROKE`, `LT SYNC`, `LT STROKE`, and `RT SYNC` cycle forever on all terrains, with frequency bounded from below and depending only on the robot structure and parameters.

The assumption we require is that the motors are strong enough to guarantee that there is a uniform upper bound $T_{\max}$ such that every position command sent to a motor is achievable within that time.

Our proof proceeds by induction along the body, starting with the first segment. The yaw states for right and left have the same structure, so without loss of generality, we will only examine `RT` states. Starting from the yaw `RT SYNC`, in all but the first segment, the yaw FSM waits until the preceding segment reaches yaw `LT STROKE`. However, in the first segment the condition is vacuously met and the yaw FSM enters the `RT STROKE` with its `WAIT RISE`, `SWING`, and `WAIT FALL` sub-states. The `WAIT RISE` and `WAIT FALL` sub-states have explicitly bounded dwell times T_{rise} and T_{fall}. The dwell time in `SWING` cannot exceed $T_{\max}$, because in any valid configuration $\psi_{\text{thresh}} < \psi_{\max}$, the yaw motor is commanded to reach $\psi_{\max}$, and the motor is strong enough to achieve this. It follows that within a uniformly bounded time of no more than $T_{\text{rise}} + T_{\text{fall}} + T_{\max}$ the first segment's yaw state machine reaches the `LT SYNC` state. Since the same logic applies to liveliness going from `LT SYNC` back to `RT SYNC`, the induction base is proven for the first segment yaw FSM: it must cycle

[5] https://purl.org/birds-lab/code/csm-CAD.

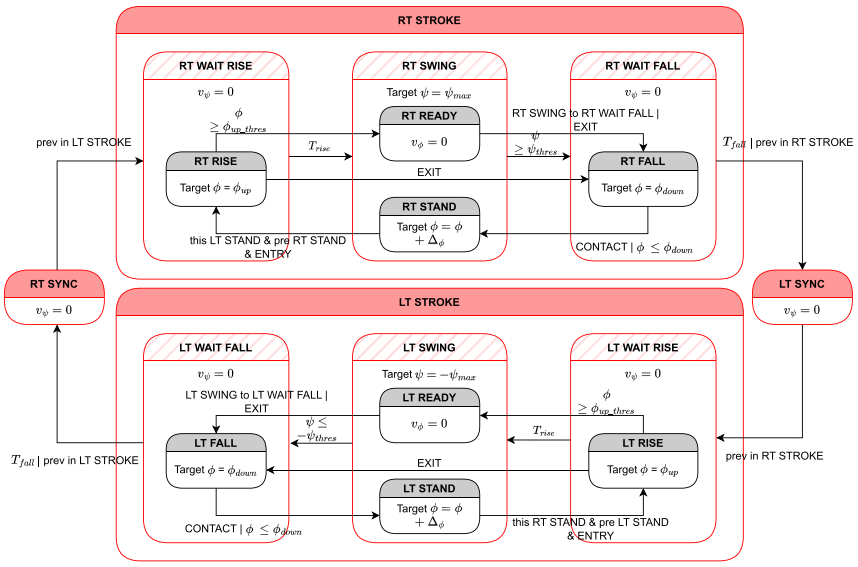


Figure 2. State machine architecture used for hierarchical yaw-leg coordination. Parameters of the FSM controller: ψ_{max}, ϕ_{up}, and ϕ_{down} define state targets; ψ_{thres} and ϕ_{up_thres} define transition thresholds; $\Delta\phi$ is the touchdown offset in `STAND`; T_{rise} and T_{fall} are dwell times in the `WAIT` states.

through within a finite time of $T_{\text{yaw}} := 2 \times (T_{\text{rise}} + T_{\text{fall}} + T_{\text{max}})$ units of time.

Consider segment $N+1$. By induction, we assume that segment N's yaw state machine must cycle within some bounded time $T_{\text{yaw}}^{[N]}$. The yaw FSM of segment $N+1$ could block in two places additional to those available to the first segment: the `SYNC` states waiting for the segment in front of it to reach the contra-lateral `STROKE` state. It follows that the segment $N+1$ yaw state machine must cycle within no more than $2T_{\text{yaw}}^{[N]} + T_{\text{yaw}}$—a finite time.

It follows that there exists a finitely bounded time such that the yaw FSM-s of all segments must cycle within that time.

All actuators used PD control with an internal trapezoidal velocity profile generator. The velocity profile generator attempted to generate the velocity trajectory with best effort, utilizing the maximum speed v_{max} during the acceleration and deceleration phases. We discovered that if the legs stop their roll motion when ground contact is detected, the height of the body ratchets down over multiple cycles, and the robot crashes into the ground.

Therefore, every time the foot reached the ground, we added a touchdown offset $\Delta\phi$ so that the feet extended downward a little further and produced a force that supports the body.

The control architecture uses several parameters listed in table 1. Although the mechanical design allowed the roll motor to have a range of $-\pi/2$ to $\pi/2$, our experiments indicated that a roll angle larger than -1.0 rad yielded the most stable walking performance. Similarly, we constrained the yaw motor range within $[-0.6, 0.6]$ to prevent the feet from interfering.

The FSM in figure 2 is parameterized by table 1. ψ_{max} defines the yaw target during `SWING`, ψ_{thres} triggers the `SWING`→`WAIT FALL` transition, ϕ_{up} and ϕ_{down} define leg roll targets in `RISE`/`FALL`, $\phi_{\text{up_thres}}$ triggers the `RISE`→`READY` transition, $\Delta\phi$ sets the touchdown offset in `STAND`, and $T_{\text{rise}}, T_{\text{fall}}$ are dwell times in the `WAIT` states.

By adjusting the parameters above, we tuned the behavior of the FSM controller. For instance, changing the maximum yaw angle (ψ_{max}) modulated the step size of each leg, affecting the walking speed; adjusting ϕ_{up}, ϕ_{down}, and $\Delta\phi$ resulted in different body heights and foot clearances, affecting the robot's ability to cross obstacles. When the robot walked on extremely rough terrain (with large variance in the vertical direction), we also increased T_{rise} and T_{fall} to give the leg enough time to rise and fall before swinging, thereby avoiding tripping or tipping over, with a small sacrifice in walking speed. Changing the motor's maximum speed had a similar, more intuitive effect on walking speed, and we did not detail those effects here.

2.3. Simulation setup

We tested the robot model and locomotion control, and collected the results using the MuJoCo simulator (version 3.3.5; figure 3). We controlled the simulated robot using the `pygame`-based `PyCKBot` library

Table 1. Parameters for tuning the control architecture.

Parameter	Definition	Range	Sim. Value	Unit
$\psi_{\max}$	Maximum yaw angle	$[-0.6, 0.6]$	0.6	radian
ψ_{thres}	Yaw threshold for yaw state transition from `SWING`→`WAIT FALL`	$(-\psi_{\max}, \psi_{\max})$	0.55	radian
ϕ_{up}	The highest point for leg roll	$[-1.0, 0]$	-0.1	radian
$\phi_{\text{up_thres}}$	Roll threshold for leg state transition from `RISE`→`READY`	$[-1.0, \phi_{\text{up}})$	-0.15	radian
ϕ_{down}	The lowest point for leg roll	$[-1.0, \phi_{\text{up_thres}})$	-1.0	radian
$\Delta\phi$	Touchdown offset	$[0, \phi_{\text{down}})$	0.6	radian
T_{rise}	`WAIT RISE` state delay	$\geqslant 0$	0.1	second
T_{fall}	`WAIT FALL` state delay	$\geqslant 0$	0.1	second

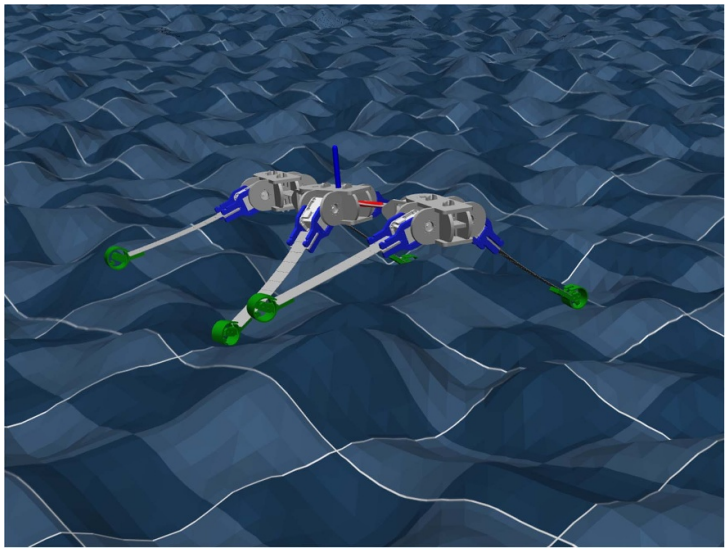


Figure 3. The 3-segment hexapod robot simulated on a rough terrain in the MuJoCo simulator.

(Revzen *et al* 2010) used for many internal projects in our lab.

We converted the robot CAD model to a compatible robot description file using the `onshape-to-robot` package and assigned the mass properties for each part in OnShape as follows. We simulated the compliant parts by dividing the part into three equal segments connected by hinge joints, with torsional stiffness chosen to produce the overall bending as specified above. We set the materials and motor characteristics to match the **Dynamixel MX-64** motors[6] and actual materials used in the physical robot (see table 2). We provided the parameters defining the FSM controller in the simulation in table 1.

3. Simulation results

We evaluated the robot's walking stability and control robustness. We defined stability through two primary lenses: (1) Synchronization, determining whether the distributed control architecture yielded a stable, rhythmic gait by synchronizing all segments under varying initial conditions; and (2) Terrain Robustness, determining whether the generated gait maintained body posture and progression across various irregular terrains.

3.1. Phase estimation and coordinate definition

Since we coupled the state machines via transition rules and they operated cyclically, they functioned as coupled oscillators, each possessing a local period and phase that contributed to a global gait cycle. To quantify synchronization, we analyzed whether the state machines achieved a phase-locked state. However, because the state machines contained static states (e.g. the `STAND` state of legs or `SYNC` state of yaw motors), traditional linear phase estimation methods like the Hilbert Transform were unsuitable. Instead, we utilized the phase estimation method developed in Revzen and Guckenheimer (2008). Specifically, we constructed a 7-dimensional signal for each segment–comprising the x and z coordinates of the foot tips, the yaw motor angle, and leg roll angles–to estimate the instantaneous phase of each segment. We calculated the phase difference as the circular difference between the circular average

[6] From https://emanual.robotis.com/docs/en/dxl/mx/mx-64/.

Table 2. Simulated robot properties.

Part	Property	Value
All 3D-printed / laser-cut parts	PLA density	1.24 g cm^3
Compliant parts	300 stainless steel density	8.0 g cm^3
Dynamixel MX-64	Mass	126 g
	No-load speed	78 RPM
	Holding torque	7.3 N·m

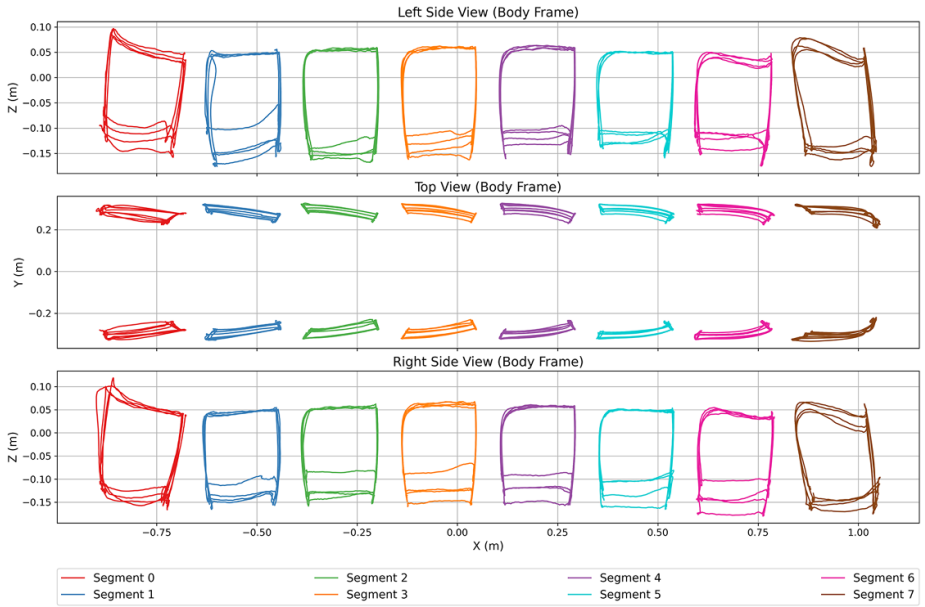


Figure 4. Toe Trajectories in body frame for 8-segment robot, on mildly rough terrain. These show the expected rectangular shape, while the bottom edge is variable, adapting to ground height.

phases of even-numbered and odd-numbered segments: $\arg\left((\sum_{k\in\text{even}}\phi_k)/(\sum_{k\in\text{odd}}\phi_k)\right)$ where the ϕ_k are phasors.

To evaluate physical stability, we analyzed the robot's body height, pitch, and roll. As the robot is a multi-segmented articulated body without a fixed torso, we defined a virtual body frame O_G:

1. Origin (O_G): We defined this as the geometric centroid of all segment positions, $O_G = \frac{1}{N}\sum_{i=0}^{N} s_i$, where s_i was the world-frame position of the ith segment.
2. Forward axis ($\hat{X_G}$): We computed this via principal component analysis. We calculated the covariance matrix $C = \sum_{i=0}^{N}(s_i - O_G)(s_i - O_G)^T$ and selected the eigenvector associated with the largest eigenvalue as $\hat{X_G}$.
3. Vertical axis ($\hat{Z_G}$): We derived this from the mean orientation of the segments. Let $\hat{z}_i$ be the local z-axis of the ith segment. We computed the mean vector $\bar{\boldsymbol{z}} = \frac{1}{N}\sum_{i=0}^{N}\hat{z}_i$ and orthonormalized it against $\hat{X_G}$ using Gram–Schmidt projection: $\mathbf{v} = \bar{\boldsymbol{z}} - (\bar{\boldsymbol{z}}\cdot\hat{X_G})\hat{X_G}$; $\hat{Z_G} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$.
4. Lateral Axis ($\hat{Y_G}$): We completed the frame via the right-hand rule: $\hat{Y_G} = \hat{Z_G} \times \hat{X_G}$.

3.2. Simulation setup

Here we report on simulations of two robot configurations—a 3-segment (6-leg) hexapod and an 8-segment (16-leg) hexadecapod—across 20 trials with randomized initial phase configurations.

In the supplemental materials we also include results for 16, 32, and 64 segments.

To contextualize the dimensions of the terrains we examined, the dimensions of the robot segments were: legs 33 [cm] hip to toe, rotating in YZ plane; segments were 33 [cm] Z motor axis to Z motor axis along X; the Y distance between right and left hips was 9 [cm].

To validate the controller's adaptability, we tested five distinct environments:

- **Floating**: We set gravity to 0 with no ground contact, isolating the intrinsic controller dynamics in a set-up reminiscent of 'fictive locomotion' (Grillner and Zangger 1979).
- **Flat**: We used a flat horizontal surface.
- **Rough**: We generated terrain using a Gaussian Process with Gaussian kernel $\sigma = 6$[cm] (i.e. 1/5 segment long), height scale 30 [cm] (i.e. approximately leg length).

- **Hill**: We generated a quadratic surface with slope ranging from $-13.3°$ to $13.3°$.
- **Stairs**: We constructed 10 upward and 10 downward steps of length 20[cm] (i.e. 2/3 segment length), height 4[cm] (i.e. 1/9 leg length).

3.3. Synchronization

A critical requirement for distributed controllers is the ability to recover a coordinated gait from arbitrary initial conditions, even in the presence of external disturbances. Figure 5 illustrates the transient response of the phase error (the deviation from the ideal alternating gait) for both robot configurations.

On Flat Terrain (left column), the mean signed phase error decayed smoothly toward zero within roughly two global cycles for every body length tested, indicating that local interactions between segments rapidly drove the system to the stable alternating limit cycle. The ± 1 s.d. band decayed in parallel and settled to a small but nonzero noise floor; the mean itself remained centered at zero, distinguishing genuine phase locking from a residual offset. On Random terrain (right column), the mean still converged to (near-)zero, with a somewhat larger noise floor reflecting irregular foot contacts and variable terrain heights. The controller did not require a perfectly flat surface to synchronize; the synchronization was robust enough to reject terrain irregularities. This behavior scaled effectively from 3 to 32 segments: although the noise-floor amplitude grew with body length, neither the convergence timescale nor the centering of the mean on zero did. At 64 segments the simulation became numerically unstable and is omitted; we attribute this to the cumulative cost of contact resolution rather than to an instability of the controller itself.

Table 3 summarizes the steady-state phase error reached at the end of these trials, reported as circular statistics (equation (2)) over all trials and segments. Across every body length and on both terrains the circular mean stayed within 0.1 rad of zero, confirming that the system locks onto the anti-phase gait rather than settling at a spurious offset; the circular standard deviation captures the residual noise floor, which grew with body length and was larger on random than on flat terrain.

3.4. Return map analysis indicates convergence

For further insights into the systems' convergence to a limiting motion cycle, we applied Poincaré return map analysis. As often done in the legged locomotion literature, we chose the center-of-mass apex event as our Poincaré section. Figures 6 and 7 show the results for the y (lateral motion), z (vertical position), and yaw (projected XY plane rotation) induced by a motion cycle relative to the body frame of the previous apex. We took these rigid body motion coordinates at 4 points segments out of the 16 in the simulated robot, since convergence for the 8-segment and 3-segment robots was so rapid as to be uninteresting. Return map plots for robots of other sizes are in the supplemental materials. The figures show a convergence onto the diagonal, indicating that each of these parameters reached a steady state.

3.5. Terrain robustness

We visualized the steady-state performance across terrains in figures SI-1 and SI-2. In the Floating condition, the controller produced a stable 'fictive locomotion' limit cycle. With no ground contact to trigger transitions, the system defaulted to a rhythmic pattern where the even–odd phase difference converged to a constant offset, which revealed the intrinsic oscillator dynamics of the FSM network.

On Flat terrain, the pattern anchored to the physical environment, producing steady forward progression with a nearly straight body path. Pitch, roll, and body height settled into small, bounded oscillations. When the robot traversed the Hill terrain[7], it maintained the anti-phase traveling wave while adapting its pitch and height to the slope. On Rough and Stair terrains[8], the benefits of the distributed architecture were most evident. While steps and local irregularities sharply deformed individual foot trajectories (visible as kinks in the foot traces), the global phase pattern remained robust. The body height tracked the terrain profile closely, and the body posture remained stable. Comparing the 3-segment and 8-segment trials, we observed qualitatively identical behavior; the 8-segment robot traversed the same obstacles with equal stability, which further validated that the proposed FSM acted as a modular, scalable pattern generator capable of navigating diverse environments.

To probe the controller's terrain adaptability across a wider range of surface properties, we ran additional simulations of the simplified 8-segment robot on flat terrains with varying friction (low: $\mu_{\text{slide}} = 0.3$; medium: 0.9; high baseline: 3.0) and on stair terrains with varying step height (half and double the default). All runs used identical controller parameters. Videos are available online:

Terrain variation	Video
Low friction ($\mu_{\text{slide}} = 0.3$)	[video link]
Medium friction ($\mu_{\text{slide}} = 0.9$)	[video link]
Low stair (half default step height)	[video link]
High stair (double default step height)	[video link]

Across this 10× range in sliding friction and 4× range in step height, the controller continued

[7] [Video] of the 8-segment robot on hill terrain.
[8] [Video] of 8-segment robot on stair terrain.

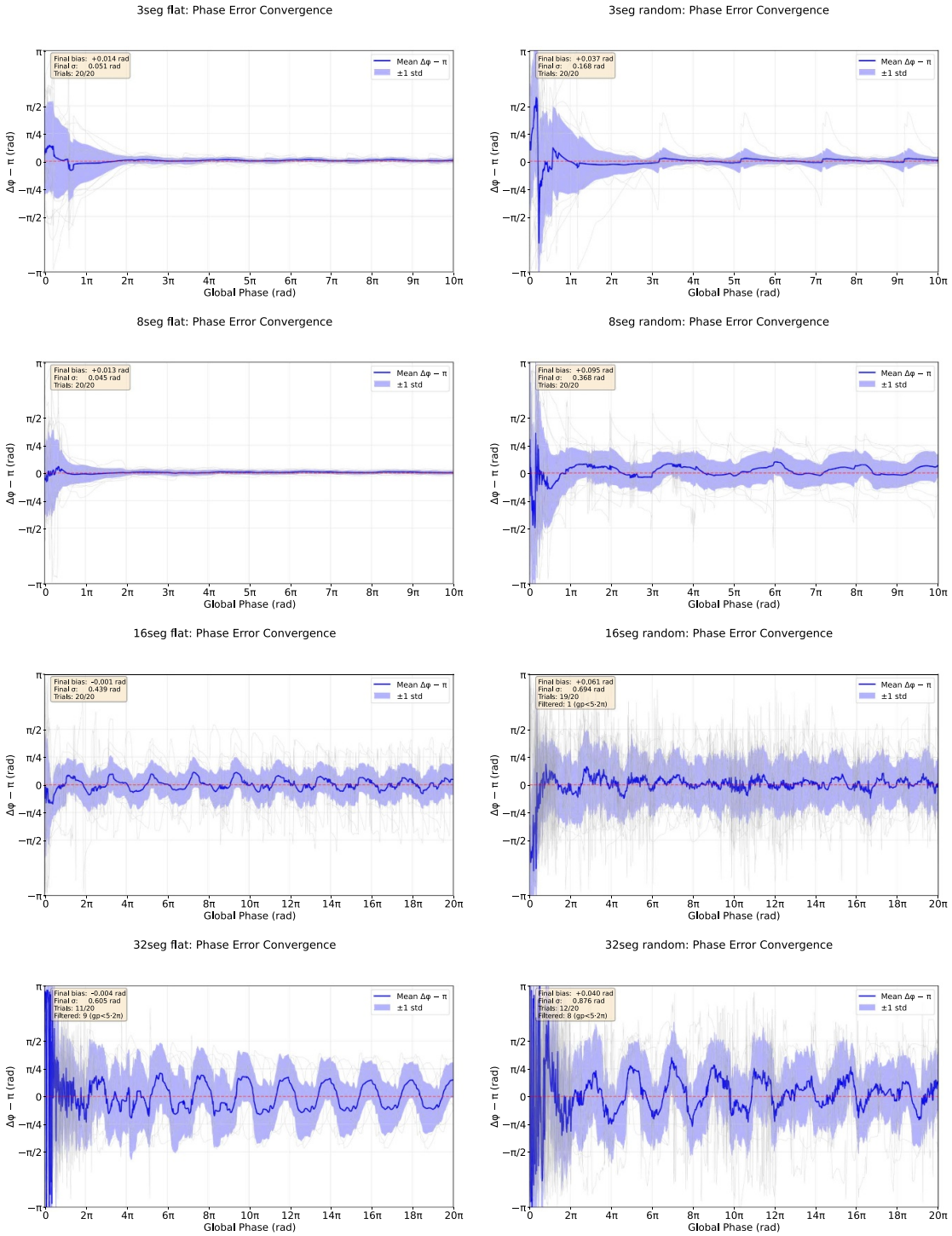


Figure 5. Robustness of phase convergence across body lengths. Each panel plots the signed phase error $\Delta\varphi - \pi$ (rad) versus global phase for 20 trials with randomized initial conditions: light gray traces are individual trials, the dark blue trace is the trial-wise circular mean, and the shaded band is ± 1 circular standard deviation. Because $\Delta\varphi - \pi$ is an angular quantity that can approach the $\pm\pi$ wraparound, the mean and spread are computed as circular statistics (equation (2)), not arithmetic averages. Rows are the 3-, 8-, 16-, and 32-segment robots; left column is ideal flat terrain, right column is random rough terrain. Plotting the signed (not absolute) error makes the mean and the spread separately interpretable: the mean converging to zero indicates phase locking onto the anti-phase gait, while the band represents the irreducible noise floor. The 64-segment configuration was numerically unstable in our simulator and is omitted. Video links of typical rough-terrain simulations: 3-segment robot [video], 8-segment robot [video].

Table 3. Steady-state phase error $\Delta\varphi - \pi$, reported as circular statistics (equation (2)) over 20 trials. Values are in radians; the circular mean stays near zero in every configuration while the circular standard deviation grows with body length and with terrain roughness.

	Flat		Random	
Robot	Mean	s.d.	Mean	s.d.
3-segment	0.014	0.051	0.037	0.168
8-segment	0.013	0.045	0.095	0.368
16-segment	−0.001	0.439	0.061	0.694
32-segment	−0.004	0.605	0.040	0.876

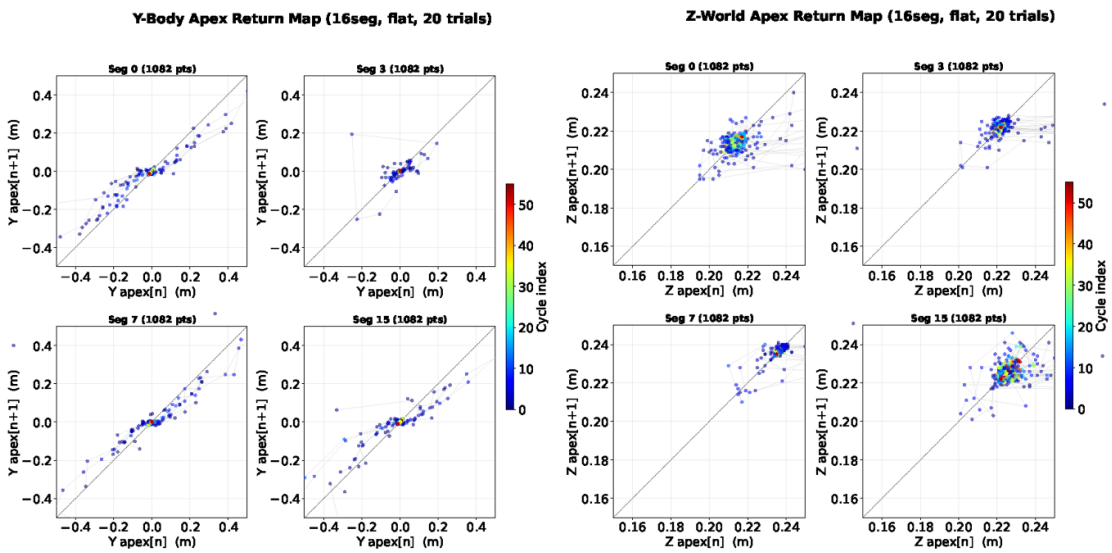


Figure 6. Apex return map of the body-frame y (left) and world-frame z (right) coordinates for the 1st, quarter-length, half-length, and final segments of the 16-segment robot, taken with respect to the apex frame of the previous apex. Points pool 20 independent trials and are colored by cycle index, so the late-cycle (warmer-colored) cluster near the diagonal is the convergent limit behavior.

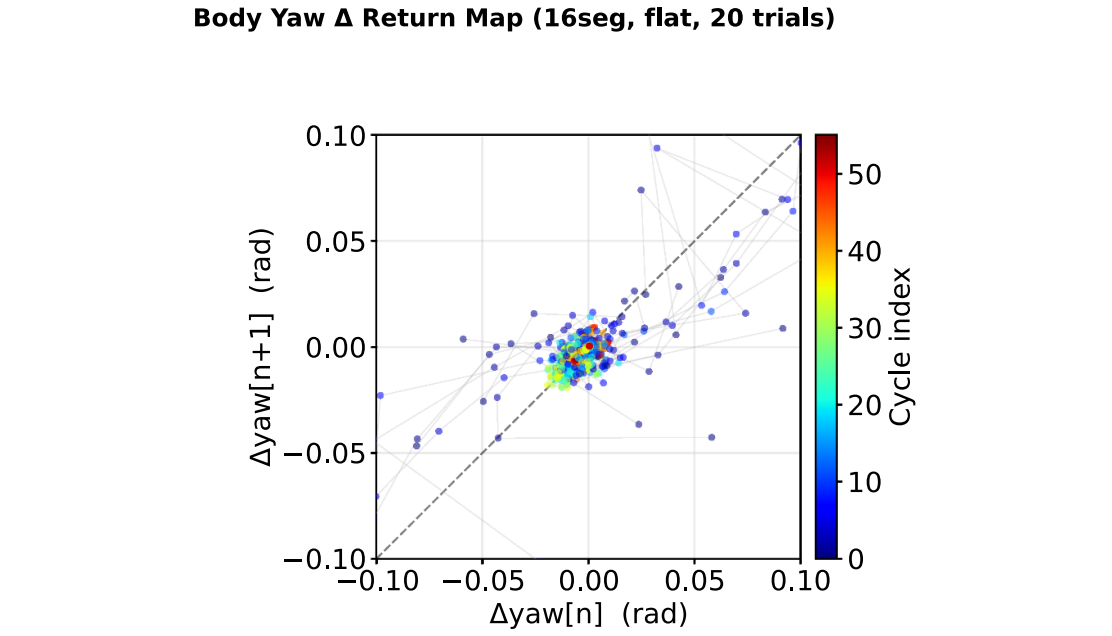


Figure 7. Apex return map of the per-cycle change in body yaw, Δyaw, for the 16-segment robot. Points pool 20 independent trials and are colored by cycle index; the tight late-cycle cluster at the origin marks the limit behavior.

to produce stable forward locomotion without any returning.

4. Discussion

The simulations presented in this paper suggested that a remarkably simple, segment-wise FSM could generate robust multi-legged locomotion over a wide range of terrains. With a single set of parameters, the controller produced a consistent traveling-wave pattern that phase-locked reliably from randomized initial conditions for 3-segment, 8-segment, 16-segment, and 32-segment simulated robots. The resulting gaits were not only stable in the sense of convergent phase relationships, but also maintained reasonably bounded body pitch, roll, and height while traversing flat ground, smooth hills, random rough terrain, and stairs. In the floating condition, the same controller exhibited fictive locomotion, which indicated that the FSM network itself embodied an intrinsic limit cycle rather than relying on carefully tuned continuous oscillators. Together, these outcomes supported the view that discrete, event-triggered pattern generators could capture much of the functionality typically associated with CPG-based designs, while remaining easy to implement, to scale, and to analyze. Crucially, unlike many dynamic models, our FSM was governed by a small number of easy to interpret parameters.

4.1. Limitations of our simulation

At the same time, our simulation study made several assumptions that limit how directly the results could be transferred to hardware. One such assumption was embedded in the bending element attaching segments to each other. We assumed it can only allow bend in the pitch direction, with no twisting around the X axis. While such torsionally stiff bending joints can be fabricated, their design is non-trivial. Torsional compliance will significantly hamper the ability of the legs to push against contra-lateral legs in the segments before and after, leading to dramatically reduced leg clearance.

Another limitation arose from our choice to switch from ZXZ 'centipede' segments of prior work to the $(XX)Zy$ design used here. The ZXZ design seemed to have a highly desirable property—its projection on the horizontal plane, under the assumption the feet are pin joints, is a four-bar linkage which has a kinematic degree of freedom. This DoF implies that the ZXZ design can walk without slipping, whereas our $(XX)Zy$ design cannot, even on flat terrain. As expected, the simulated feet exhibited slipping which was particularly noticeable on flat and gently sloped terrain.

Humans and other large land mammals are highly averse to slipping, and we tend to project this as a design requirement for our robots. It has been shown that (1) allowing slipping can dramatically improve the kinematic operational envelope Zhao and Revzen (2020); (2) multi-legged animals such as cockroaches are constantly slipping as they move Sachdeva *et al* (2018); and (3) the presence of dynamic coulomb friction slipping does not alter the principally kinematic nature of the motion Zhao *et al* (2022). Our results here showed that despite all the slipping, the robot still progressed forward reliably across all tested terrains with the current body plan and controller, and the systematic presence of slipping did not adversely affect the ability to move. It is unclear what, if any, is the kinematic advantage of non-slip motion, whereas the advantage of requiring fewer actuators is obvious. From a design perspective, the common approach of using three motors per leg requires about twice as many actuators as our design which requires $3N-1$ motors for $2N$ legs. There were also more conventional places the model was unrealistic. We modeled friction as isotropic and time-invariant, whereas real surfaces exhibit spatial variation, wear, and contamination. The rough terrain was still a rigid and smooth 2.5 dimensional topography, unlike real gravel and soil which flow and crumble, and have overhangs. The controller itself did not incorporate explicit foothold planning, perception, or online adaptation; it reacted only through local contact information and the simple touchdown-offset rule.

4.2 There are no 'periodic gaits' on random terrains

It is mathematically natural to ask whether our control method produces a periodic gait, but this mathematically natural question is a distraction from understanding animal gaits and controlling robots effectively. The rigid mathematical structure of strict periodicity does not reflect what animals actually do. At the very least animal exhibit motions that drift within 'gait families', i.e. parametrically related collection of related periodic gaits such as human running gaits over a range of speeds. However, the moment noise—the uncertainty in the physical world—starts affecting the trajectory, mathematical periodicity become vanishingly unlikely. For systems with stochastically perturbed time evolution, there is no clearly accepted extension of the notion of periodicity. The recent Wilshin *et al* (2025) offers one approach, by defining phase in a stochastic system, and providing statistically reliable means for its estimation.

Some groups have investigated open-loop periodic control schemes. In these schemes, the control signals in the robot are periodic and oblivious to

the environment, offering the possibility that entrainment of the dynamics to the driving oscillator in the robot would create a periodic gait. One such robot was RHex, which showed excellent results on rough terrain with its open loop controller (Saranli *et al* 2001). The Sastra *et al* (2008) 'centipede' and the Zhao (2021) 'multipod' robots which inspired our hardware design were similarly open loop, as were the robots in Chong *et al* (2023). Unfortunately, even for RHex, the simplest of these robots, the closed loop, robot in environment dynamics may well be chaotic dynamics which are only periodic in some averaged sense (Seipel and Holmes 2007, figures 8–10).

Our proposed FSM controller is closely related to prior studies on multi-legged robot locomotion and biological walking, and in some senses is a return to the pioneering work of McGhee (1967). Biological studies suggest that the gait patterns of multi-legged animals do not necessarily converge to a single fixed pattern. For example, both Yasui *et al* (2022) and Cruse *et al* (1998) proposed decentralized control models in which each leg performs state transitions based on local contact information and the states of neighboring limbs, enabling adaptive walking behavior on irregular terrain, and producing periodic-seeming locomotion on flat ground.

Our results here provide a different approach—instead of periodicity, we have proven 'liveliness' (see section 2.2.1), which is a looser term that is often used in the analysis of real-time software systems. Finite state machine models lend themselves naturally to both manual and automated proofs of liveliness. We have been unable to prove liveliness of the controller presented here in its entirety. Our proof applies only to the yaw state machines, allowing for the leg oscillators to get stuck. Also, the liveliness bounds we obtained become weaker the further along the robot the segment is.

Nonetheless, toe motions in the body frame seem fairly regular, varying only in stance to adapt to terrain height (see figure 4). Examining stability through the lens of the returns maps of various quantities, apex maps of the 16 segment robot show clear signs of convergence onto some limit behavior (see figures 6 and 7). When examining the same type of plots for longer robots, slow convergence or perhaps even oscillations seem to appear (see supplemental figures SI-7 – SI-9).

4.3. Future work

Looking forward, several directions appear promising.

Much work remains on building a physical robot. This includes installing a backbone y flexure with high torsional X-axis stiffness, which may be a critical step in designing for rougher terrain. We also plan to replace the contact-switch based ground contact sensing with a strain gauge on the legs, which could make ground contact detection much more robust to mechanical damage.

On the control side, the FSM presented here could be modified to improve its provable liveliness properties. It could then serve as a scaffold for more sophisticated behaviors. For example, higher-level control modules could modulate state transition thresholds or desired joint ranges based on sensory feedback, or learning-based policies could be layered on top to adjust parameters online while preserving the underlying phase-locked structure. We could also incorporate additional feedback. An IMU might help with detecting large pitch and roll excursions, and perhaps allow the robot to handle larger slopes by feedback. A magnetometer or other absolute heading sensor might allow the robot to maintain its overall heading on very rough terrain.

Further simulation studies may also be in order. Some topics for such investigations include:

(1) study and mitigation of the yaw oscillations that appear at the hind parts of very long robots;

(2) study of the limitations of this architecture in terms of terrain roughness, slope, etc;

(3) examination of the relationship between motor capabilities and the performance envelope of the simulated robot.

Overall, while the present work was limited by idealized models and simplified mechanics, it demonstrated that a modular FSM controller combined with appropriately designed compliant structures could provide a robust and scalable bio-inspired foundation for legged locomotion over challenging terrain.

Acknowledgment

The authors would like to thank Yifan (Ivan) Jing for his help with the 32 and 64 segment simulations. We also thank Federico Seghizzi and Marion Anderson for their help in building the earlier versions of the physical robot. This work was funded in part by NSF CPS 2038432.

Data availability statement

The data that support the findings of this study will be openly available following an embargo at the following URL/DOI:. https://purl.org/birds-lab/code/modular-fsm(ioppcite). Data will be available from 24 April 2026.

Supplementary data 1 available at: https://doi.org/10.1088/1748-3190/ae9261/data1.

Supplementary data 2 available at: https://doi.org/10.1088/1748-3190/ae9261/data2.

Supplementary data 3 available at: https://doi.org/10.1088/1748-3190/ae9261/data3.

Supplementary data 4 available at: https://doi.org/10.1088/1748-3190/ae9261/data4.

Supplementary data 5 available at: https://doi.org/10.1088/1748-3190/ae9261/data5.

Supplementary data 6 available at: https://doi.org/10.1088/1748-3190/ae9261/data6.

Supplementary data 7 available at: https://doi.org/10.1088/1748-3190/ae9261/data7.

Supplementary data 8 available at: https://doi.org/10.1088/1748-3190/ae9261/data8.

Supplementary data 9 available at: https://doi.org/10.1088/1748-3190/ae9261/data9.

Supplementary data 10 available at: https://doi.org/10.1088/1748-3190/ae9261/data10.

Author contributions

Zhuoyang Chen 0009-0009-3968-516X
Investigation (lead), Software (lead), Writing – original draft (equal)

Xinyuan Wang 0009-0002-6809-3584
Writing – original draft (supporting)

Shai Revzen 0000-0002-2989-0356
Conceptualization (lead), Funding acquisition (lead), Supervision (lead), Writing – original draft (lead)

References

Altendorfer R, Moore N, Komsuoglu H, Buehler M, Brown H B, McMordie D, Robert U S Full J and Daniel E K 2001 Rhex: a biologically inspired hexapod runner *Auton. Robots* **11** 207–13

Arena P, Fortuna L, Frasca M and Sicurella G 2004 An adaptive, self-organizing dynamical system for hierarchical control of bio-inspired locomotion *IEEE Trans, Syst. Man Cybern. B* **34** 1823–37

Autumn K *et al* 2005 Robotics in scansorial environments *Proc. SPIE* **5804** 291–302

Available at: https://purl.org/birds-lab/code/modular-fsm

Bellegarda G and Ijspeert A 2022 Cpg-rl: learning central pattern generators for quadruped locomotion *IEEE Robot. Autom. Lett.* **7** 11235–42

Bidaye S S, Bockemühl T and Büschges A 2018 Six-legged walking in insects: how cpgs, peripheral feedback and descending signals generate coordinated and adaptive motor rhythms *J. Neurophysiol.* **119** 691–706

Blickhan R and Full R J 1993 Similarity in multilegged locomotion: bouncing like a monopode *J. Comp. Physiol. A* **173** 509–17

Campbell D and Buehler M 2003 Stair descent in the simple hexapod 'rhex *Proc. IEEE Int. Conf. on Robotics and Automation* (https://doi.org/10.1109/ROBOT.2003.1241784)

Cavagna G A, Heglund N C and Taylor C R 1977 Mechanical work in terrestrial locomotion: two basic mechanisms for minimizing energy expenditure *Am, J. Physiol,-Regul. Integr. Comp. Physiol.* **233** R243–61

Chong B, Juntao H, Shengkai Li, Erickson E, Diaz K, Wang T and Goldman D I 2023 Self-propulsion via slipping: frictional swimming in multilegged locomotors *Proc. Natl Acad. Sci.* **120** e2213698120

Crespi A and Ijspeert A 2008 Online optimization of swimming and crawling in an amphibious snake robot *IEEE Trans. Robot.* **24** 75–87

Cruse H, Kindermann T, Schumm M, Dean J and Schmitz J 1998 Walknet—a biologically inspired network to control six-legged walking *Neural Netw.* **11** 1435–47

Deshpande A M, Hurd E, Minai A A and Kumar M 2023 Deepcpg policies for robot locomotion *IEEE Trans, Cogn. Dev. Syst.* **15** 2108–21

Dickinson M H, Farley C T, Full R J, Koehl M A, Kram R and Lehman S 2000 How animals move: an integrative view *Science* **288** 100–6

Edgerton V R, Grillner S, Sjöström A and Zangger P 1976 Central generation of locomotion in vertebrates *Neural Control of Locomotion* (Springer) (https://doi.org/10.1007/978-1-4757-0964-3%5F18)

Grillner S and Zangger P 1979 On the central generation of locomotion in the low spinal cat *Exp. Brain Res.* **34** 241–61

Haynes G C and Rizzi A A 2006 Gait regulation and feedback on a robotic climbing hexapod *Robot.: Sci. Syst.* **16** 97–104

Holmes P, Full R J, Koditschek D and Guckenheimer J 2006 Dynamics of legged locomotion: models, analyses and challenges *SIAM Rev.* **48** 207–304

Hultborn H and Nielsen J B 2006 Spinal control of locomotion–from cat to man *Acta Physiol.* **189** 111–21

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

Ijspeert A J and Crespi A 2007 Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model *Proc. IEEE Int. Conf. on Robotics and Automation* (https://doi.org/10.1109/ROBOT.2007.363797)

Inagaki S, Yuasa H and Arai T 2003 Cpg model for autonomous decentralized multi-legged robot system *Robot. Auton. Syst.* **44** 171–9

Klaassen B, Linnemann R, Spenneberg D and Kirchner F 2002 Biomimetic walking robot scorpion: control and modeling *Auton. Robots* **41** 69–76

Lee J, Hwangbo J, Wellhausen L, Koltun V and Hutter M 2020 Learning quadrupedal locomotion over challenging terrain *Sci. Robot.* **5** 6992–8

Lin Y, Tian Y, Xue Y, Han S, Zhang H and Lai W 2021 Innovative design and simulation of a transformable robot with flexibility and versatility, rhex-t3 *Proc. IEEE Int. Conf. on Robotics and Automation* (https://doi.org/10.1109/ICRA48506.2021.9561060)

Liu X, Gasoto R, Jiang Z, Onal C and Jie F 2020 Learning to locomote with artificial neural-network and cpg-based control in a soft snake robot *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (https://doi.org/10.1109/IROS45743.2020.9340763)

Marder E and Bucher D 2001 Central pattern generators and the control of rhythmic movements *Curr. Biol.* **11** R986–96

McGhee R B 1967 Finite state control of quadruped locomotion *Simulation* **9** 135–40

Nelson G M, Quinn R D, Bachmann R J, Flannigan W C, Ritzmann R E and Watson J T 1997 Design and simulation of a cockroach-like hexapod robot *Proc. IEEE Int. Conf. on Robotics and Automation* (https://doi.org/10.1109/ROBOT.1997.614284)

Neveln I D, Tirumalai A and Sponberg S 2019 Information-based centralization of locomotion in animals and robots *Nat. Commun.* **10** 3655

Okhotsimski D E and Platonov A K 1973 Control algorithm of the walker climbing over obstacles *Proc. 3rd Int. Joint Conf. on Artificial Intelligence* (available at: www.ijcai.org/Proceedings/73/Papers/033B.pdf)

Pearson K G 1972 Central programming and reflex control of walking in the cockroach *J. Exp. Biol.* **56** 173–93

Revzen S and Guckenheimer J M 2008 Estimating the phase of synchronized oscillators *Phys. Rev. E* **78** 051907

Righetti L and Ijspeert A J 2008 Pattern generators with sensory feedback for the control of quadruped locomotion *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA) (Pasadena, CA, USA)* IEEE (https://doi.org/10.1109/ROBOT.2008.4543306)

Roennau A, Heppner G, Pfotzer L and Dillmann R 2013 Lauron v: Optimized leg configuration for the design of a bio-inspired walking robot *Climbing and Walking Robots* (World Scientific) (https://doi.org/10.1142/9789814525534%5F0071)

Rutishauser S, Sprowitz A, Righetti L and Ijspeert A J 2008 Passive compliant quadruped robot using central pattern generators for locomotion control *Proc. IEEE RAS & EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)* (https://doi.org/10.1109/BIOROB.2008.4762878)

Sachdeva V, Zhao D and Revzen S 2018 Cockroaches always slip a lot *Integrative and Comparative Biology* vol 58 (OXFORD UNIV PRESS INC JOURNALS DEPT, 2001 EVANS RD, CARY, NC 27513 USA) pp E412–412

Saranli U, Buehler M and Koditschek D E 2001 Rhex: a simple and highly mobile hexapod robot *Int. J. Robot. Res.* **20** 616–31

Saranli U, Rizzi A A and Koditschek D E 2004 Model-based dynamic self-righting maneuvers for a hexapedal robot *Int. J. Robot. Res.* **23** 903–18

Sartoretti G, Shaw S, Lam K, Fan N, Travers M and Choset H 2018 Central pattern generator with inertial feedback for stable locomotion and climbing in unstructured terrain *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)* (https://doi.org/10.1109/ICRA.2018.8461013)

Sastra J, Heredia W B, Clark J E and Yim M 2008 A biologically-inspired dynamic legged locomotion with a modular reconfigurable robot *Proc. ASME Dynamic Systems and Control Conf.* pp 1467–74

Sastra J, Revzen S and Yim M 2012 Softer legs allow a modular hexapod to run faster *Adaptive Mobile Robotics* (World Scientific) pp 507–10

Schilling M, Hoinville T, Schmitz J and Cruse H 2013 Walknet, a bio-inspired controller for hexapod walking *Biol. Cybern.* **107** 397–419

Seipel J and Holmes P 2007 A simple model for clock-actuated legged locomotion *Regul. Chaotic Dyn.* **12** 502–20

Shai Revzen J S, Eckenstein N and Yim M 2010 Ckbot platform for the icra 2010 planetary challenge *Proc. of IEEE Int. Conf. on Robotics and Automation, Workshop 'Modular Robots: The State of the Art* (IEEE) pp 11–12

Sprowitz A, Kuechler L, Tuleu A, Ajallooeian M, Möckel R and Ijspeert A J 2011 Oncilla robot: a light-weight bio-inspired quadruped robot for fast locomotion in rough terrain *Proc. 5th Int. Symp. on Adaptive Motion of Animals and Machines (AMAM)* EPFL-CONF-182313

Veskos P and Demiris Y 2005 Developmental acquisition of entrainment skills in robot swinging using van der pol oscillators *Epigenetic Robotics Workshop* (https://doi.org/10.1142/97898144159 58%5F0065)

Wilshin S, Kvalheim M D, Scott C and Revzen S 2025 Estimating phase from observed trajectories using the temporal 1-form *Neural Comput.* **37** 2158–204

Wilson D M 1961 The central nervous control of flight in a locust *J. Exp. Biol.* **38** 471–90

Yasui K, Takano S, Kano T and Ishiguro A 2022 Adaptive centipede walking via synergetic coupling between decentralized control and flexible body dynamics *Front. Robot. AI* **9** 797566

Zhang J, Liu Q, Zhou J and Song A 2022 Crab-inspired compliant leg design method for adaptive locomotion of a multi-legged robot *Bioinsp. Biomim.* **17** 025001

Zhang Z, Bellegarda G, Shafiee M and Ijspeert A 2024 Online optimization of central pattern generators for quadruped locomotion *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (https://doi.org/10.1109/IROS58592.2024.10802762)

Zhao D 2021 Locomotion of low-dof multi-legged robots *PhD Thesis* University of Michigan

Zhao D, Bittner B, Clifton G, Gravish N and Revzen S 2022 Walking is like slithering: a unifying, data-driven view of locomotion *Proc. Natl Acad. Sci.* **119** e2113222119

Zhao D and Revzen S 2020 Multi-legged steering and slipping with low dof hexapod robots *Bioinsp. Biomim.* **15** 045001