Learning Neuroplastic Matching of Robot Dynamics in Closed-loop CPGs

Felix Ruppert (ruppert@is.mpg.de)
Max Planck Institute for Intelligent Systems https://orcid.org/0000-0002-4200-1250
Alexander Badri-Spröwitz
Max Planck Institute for Intelligent Systems

Article

Keywords: robots, control task dynamics, central pattern generator

Posted Date: November 22nd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1016591/v1

Version of Record: A version of this preprint was published at Nature Machine Intelligence on July 18th, 2022. See the published version at https://doi.org/10.1038/s42256-022-00505-4.

Learning Neuroplastic Matching of Robot Dynamics in Closed-loop CPGs

Felix Ruppert and Alexander Badri-Spröwitz

Abstract— **Legged robots have the potential to show locomotion performance with reduced control effort and energy efficiency by leveraging elastic structures inspired by animals' elastic tendons and muscles. However, it remains a challenge to match the natural dynamics of complex legged robots and their control task dynamics. Here we present a framework to match control task dynamics and natural dynamics based on the neuroelasticity and neuroplasticity concept. Inspired by animals we design quadruped robot Morti with strong natural dynamics as a testing platform. It is controlled through a bioinspired closed-loop central pattern generator (CPG) that is designed to neuroelastically mitigate short term perturbations using sparse contact feedback. We use the amount of neuroelastic activity as a proxy to quantify the dynamics' mismatching. By minimizing neuroelastic activity, we neuroplastically match the control task dynamics to the robot's natural dynamics. Through matching the robot learns to walk within one hour with only sparse feedback and improves its energy efficiency without explicitly minimizing it in the cost function.**

Animals can locomote with grace and efficiency that is superior to legged robots. Because of elastic mechanisms in their leg design animals can safely traverse rough and unstructured terrain [1], [2]. These elastic mechanisms mitigate the interactions between walking systems and the environment. The interactions are nonlinear and non-continuous [3] and are defined by switching dynamics and a high degree of uncertainty [4].
To achieve similar locomotion behavior roboticists aim to design bioinspired robots that utilize passive elastic structures that provide the same advantages to robots and simplify the control task [1], [5]–[7]. By designing mechanical properties like impedance [8]–[11] and spring-loaded inverted pendulum behavior (SLIP) [2], [12], [13], the natural dynamics can be designed so the control system can exploit the system's mechanics, to achieve viable behavior with no or reduced control effort, improved energy efficiency and robustness [14]–[16]. A robot with a mechanical spring in the leg design will bounce passively with fairly simple control algorithms [17], [18]. Opposite to that, a fully actuated robot, with strong motors and less prominent passive natural dynamics needs constantly active and more sophisticated control algorithms to achieve similar hopping behavior [19]–[21].

In a system with strong natural dynamics, we aim to design the control task dynamics so they **match** the natural dynamics and the controller can leverage the passive elements. If the dynamics do not match, the controller requires additional energy to enforce a desired behavior (see Toy example section S5). Yet so far, no formulation for the matching of the control task to a given robot's dynamics exists, especially for robots with engineered natural dynamics. Previous work focused on designing specific aspects of natural dynamics to fit a given control scheme [22]–[24]. In this work, we focus on quantifying the match between control task dynamics and natural dynamics, and how to improve and learn matching.

In animals the neural structure and neuromuscular pathways evolved over many generations and are inherent to each individual at birth [25]. In robotics the control laws and electrical connections are hardcoded in the design phase before production. The timing and intensity of muscle activity patterns in animals and robot motor controller activity however, have to be matched to the system's natural dynamics as a lifelong learning task in animals [26], [27] or during the development of a robot.

In this study, we implement a quadrupedal robot with engineered natural dynamics that is controlled by a central pattern generator (CPG). CPGs are neural networks found in animals that produce rhythmic output signals from non-rhythmic inputs [28], [29] for tasks like chewing, breathing, and legged locomotion [30], [31]. In robotics, CPGs are used as joint trajectory generators [9], [32], [33], or bio-inspired muscle activation pattern generators [34], [35]. Feedforward CPGs dictate control and coordination of motor or muscle activation without knowledge of the system's dynamics. These model-free feedforward patterns work well in combination with passive elastic leg designs that provide passive stability and robustness [9], [34], [35]. By closing feedback loops in CPGs, the system can actively react to unforeseen influences from its environment and mitigate perturbations [29], [32], [36] like unstructured terrain.
In our quadruped robot, we implement feedback and reflexes and observe the robot's behavior measured through sparse feedback from contact sensors on the robot's feet. This neuroelastic activity aims to correct discrepancies between desired and measured robot behavior (Figure 2a). We transfer the concept of **neuroelasticity** from neuroscience that describes the handling of stochastic short-term perturbations while interacting with the environment[37] into locomotion control. To quantify the matching of the robot's natural dynamics and the control task dynamics we use the neuroelastic activity as a proxy. If the dynamics do not

Dynamic Locomotion Group, Max Planck Institute for Intelligent Systems, Stuttgart, Germany
Correspondance: ruppert@is.mpg.de

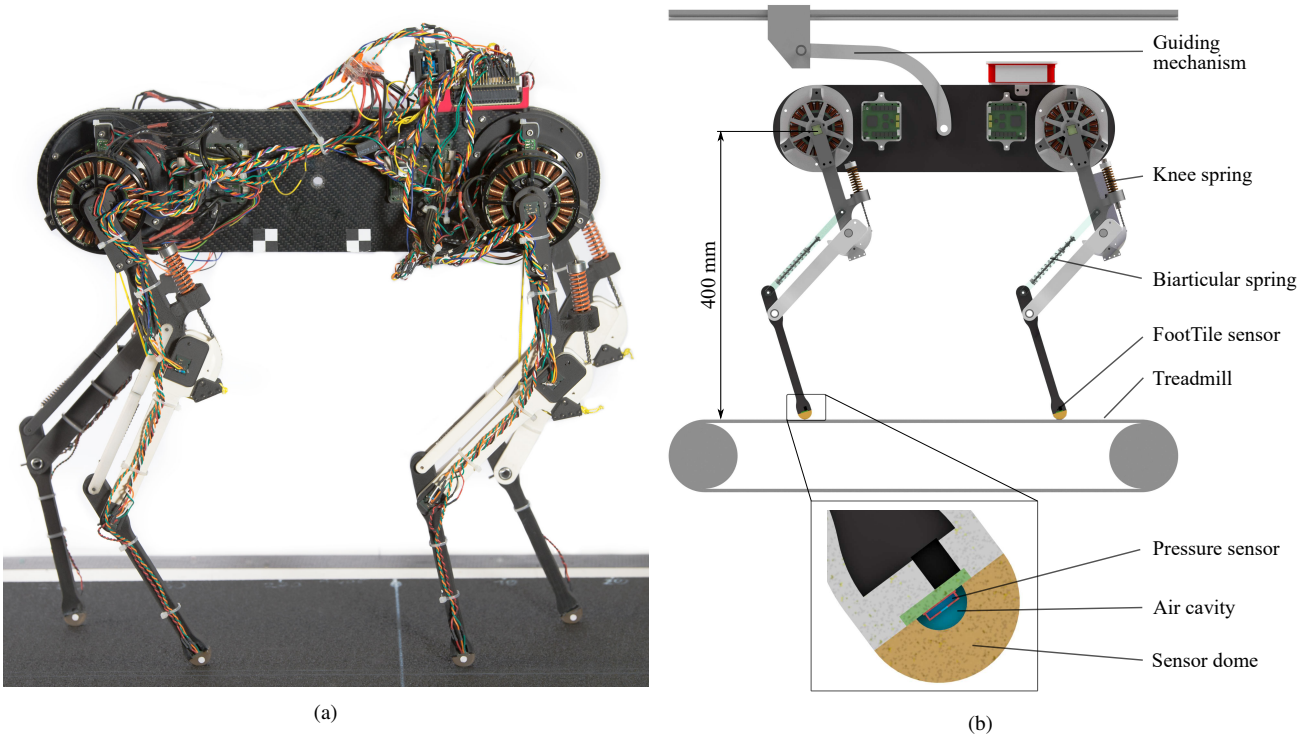


Fig. 1: **Quadruped robot Morti. a,** Photo of quadruped robot Morti. **b,** Render of the robot on top of the treadmill. The robot is constrained to the sagittal plane by a linear rail and lever mechanism that allows body pitch around the robot's center of mass. Contact of the robot is measured by four FootTile contact sensors. Cutout view: Cut view of the FootTile sensor on the foot segment of the robot. Under load the polyurethane sensor dome deforms and the pressure sensor measures the increasing pressure in the air cavity to detect foot contact.

match, the feedback mechanisms constantly have to intervene to correct for the discrepancy between the commanded and measured behavior of the robot.

We optimize the CPG parameters that describe the control task dynamics by minimizing the amount of neuroelastic activity. By minimizing the amount of neuroelastic activity we neuroplastically improve the matching between the natural dynamics and control task dynamics. **Neuroplasticity** describes the non-reversible effects environmental stimuli have on neural development, or more general it describes adapting the system's long term behavior to environmental requirements [37]. While neuroelastic activity in robotics act in the range $\ll 100\,\mathrm{Hz}$ [38]–[40], neuroplasticity, as a long-term learning process acts in the order of minutes to years.

To optimize and tune the control tasks, different methods have been used such as optimization [41]–[43], self-modeling [44], adaptive CPGs [45]–[47], and machine learning techniques [21], [47]–[52]. For this study, we apply Bayesian optimization [53], [54] to minimize the amount of neuroelastic activity to neuroplastically adapt the CPG parameters.

In previous work, Owaki et al. [55] presented a control approach that shows spontaneous gait transition based on mechanical coupling ('physical communication'). The CPG is coupled through mechanical coupling. Buchli et al. [47] presented an adaptive oscillator that adapts its frequency to the natural frequency of a SLIP-like simulation model. In their adaptive frequency oscillator approach the matching of control frequency and natural frequency leads to performance improvements and reduction in energy requirements. Fukuoka et al. [45] implemented short-term reflexes that adapt the robot's controller to the motion of the robot induced by external perturbations. Through a closed-loop CPG that incorporates the 'rolling body motion' the robot can actively adapt to its surrounding. Thandiackal et al. [35] showed, that feedback from hydrodynamic pressure in CPGs can lead to self-organized undulatory swimming. Yet so far no approaches for long-term neuroplastic matching of control task dynamics and complex walking system's natural dynamics in passive elastic robots exist.

To reduce the experimentation cost in terms of wear, critical failure and time, the control task dynamics are first optimized on a simulated robot. After successful optimization in simulation, the acquired optimal parameter set is applied in hardware. We transfer the optimized CPG parameter set into hardware to measure the performance of the real robot and validate the effectiveness of our approach by evaluating a performance measure.

While optimization and learning in simulation are efficient and cheap, the transfer of control policies can be difficult due to the sim2real gap [20], [21], [52]. We examine the transferability of our approach by quantifying the sim2real gap comparing simulation and hardware experiments.

The novelty of this study is twofold:

- We design reflex-like neuroelastic CPG feedback pathways triggered by foot contact. We measure neuroelastic

activity as a proxy for the mismatching between the robot's natural dynamics and the control task dynamics.
- We neuroplastically minimize the required neuroelastic activity through model-free Bayesian optimization. We achieve improved dynamics matching that enables the robot to walk within one hour. Our approach improves energy efficiency without explicit formulation in the cost function. The improved energy efficiency is evidence for increased matching.

I. Results

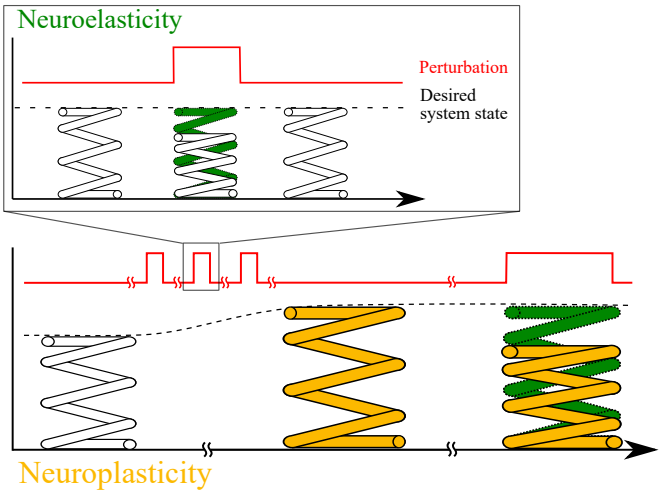
We first examine the performance of the feedback mechanisms in simulation (Figure 4b). The feedback mechanism for late touchdown (r_{LTD}), shown in red, decelerates the phase of the front left leg to wait until ground contact is established. The deceleration is visible in the flatter gradient of the oscillator phase when the mechanism is active.
The early touchdown mechanism (r_{ETD}) triggers a knee pull-up reflex (purple line) to shorten the leg to prevent further impact. In the event of early toeoff (r_{ETO}) shown in yellow, the knee flexion starts earlier than instructed by the feed-forward CPG. The late toeoff mechanism (r_{LTO}) measures the mismatching of control task and natural dynamics but does not trigger a feedback mechanism. The feedback mechanisms help the robot to mitigate perturbations stemming from dynamics mismatching. This mitigation effect is especially important in the first rollouts of the optimization where good dynamics matching is not yet achieved.
In this rollout, the late touchdown mechanism is active for 7% of the step cycle, the early touchdown mechanism is active for 5% of the step cycle, the late toeoff mechanism is active for 8% of the step cycle and the early toeoff mechanism is active for 9% of the step cycle.

We find that 150 rollouts in simulation (Figure 5a) are sufficient to achieve good locomotion performance as measured by our cost function. Each rollout in simulation took an average of 23 s for 20 s of simulation runtime on an Intel i7 CPU, making the whole optimization duration roughly one hour. The hardware rollouts were roughly one minute long to ensure stable locomotion. When the robot reaches stable behavior, 10 s are evaluated like in the simulation rollouts.
After initialization of the Gaussian kernel with 15 rollouts with random CPG parameters, the optimizer starts to approximate the reward function and performance converges towards the optimum point.

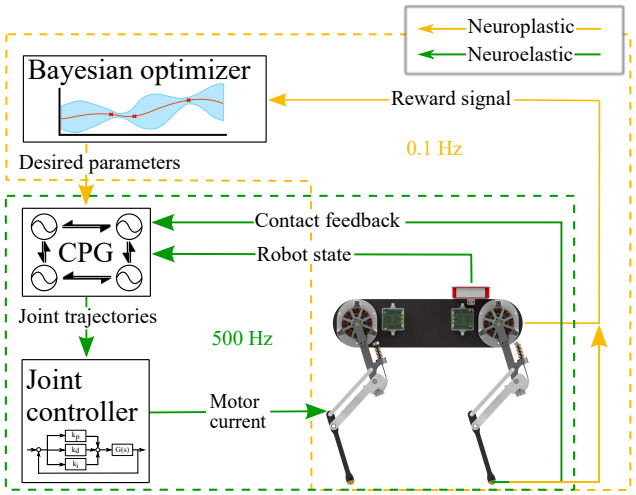
During the whole optimization, the robot fell 16 times or 11% of rollouts. 9 of the failed rollouts occurred during the first 15 rollouts with random CPG parameters.
Through optimization, the simulated robot increases its performance from the least-performing rollout (rollout 107, reward 5.62) to the optimal rollout (rollout 109, reward 2.59) by 215%. In comparison, the simulation results transferred to hardware score a reward between 5.65 and 4.41. The mean simulation reward is 3.49$\pm$ 0.66, the median simulation reward is 3.34. The mean hardware reward is 4.96 $\pm$ 0.38. The best simulation result is 41% lower compared to the lowest hardware result.
To validate the performance as well as the differences between simulation and hardware rollouts in detail we investigate the individual reward factors (Table S3) for both simulation and hardware rollouts (Figure 5a). We find that no one reward factor is responsible for the higher returned reward. Rather, all reward factors are slightly higher and their summation leads to the higher reward returned for the hardware results. The distance reward term (J_{distance}) and the feedback reward term (J_{feedback}) attribute the highest difference between simulation and hardware reward value. J_{distance} has a mean hardware reward of 2.13 $\pm$ 0.36 compared to a simulation reward of 1.67. J_{feedback} has a mean hardware reward of 0.43 $\pm$ 0.06 compared to the simulation reward of 0.13. We assume the difference is due to modeling assumptions that were made in the simulation. The hardware robot shows a lower speed due to contact losses, gearbox backlash, friction and elasticity in the FootTile sensors. During touchdown, imperfect contact of the feet leads to higher feedback activity which is penalized by the feedback reward term. The body pitch reward term J_{pitch} is in the range of the simulated reward, the mean hardware reward is 0.85 $\pm$ 0.22 compared to 0.80 in simulation. Both during the optimization shown here and initial tests the simulated robot showed more body pitch for untuned CPG parameters and the robot flipped over during several rollouts. This never happened on the hardware and even in early experiments the hardware robot never pitched more than 30°.
The periodicity reward term ($J_{\mathrm{periodicity}}$) (hardware: 0.15 $\pm$ 0.33, simulation: 0.0) and the contact reward term (J_{contact}) (hardware: 0.12 $\pm$ 0.09, simulation: 0.03) behave similarly in simulation and hardware rollouts. The similarity is expected since both simulation and hardware gaits converge to the desired gait and the latter three reward terms were introduced to guide the optimizer in finding natural gaits mostly during the first rollouts.
At the core of our approach, we hypothesize that matching dynamics improves energy efficiency. We therefor explicitly do not incorporate energy efficiency into the cost function. To quantify how matching dynamics improves energy efficiency we calculate a normalized torque as a measure of performance:

$$\tau_{normal} = \sum_{n=1}^{4} \frac{(\overline{\tau}_{\mathrm{hip,n}} + \overline{\tau}_{\mathrm{knee,n}})}{\overline{v}_{\mathrm{body}}} \qquad (1)$$

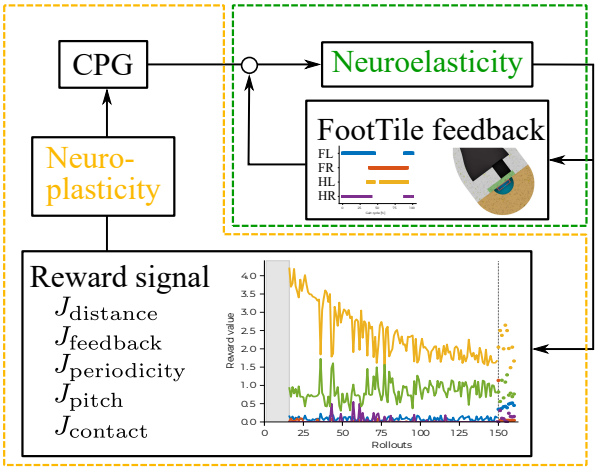
where n is the leg index, τ_{knee} and τ_{hip} are the mean knee and hip torque per rollout per leg and $\overline{v}_{\mathrm{body}}$ is the mean body velocity of the respective rollout.
The initial normalized torque is 2.52, the final value is 1.02. The mean normalized torque is 1.7$\pm$0.5, the median normalized torque is 1.55 (Figure 6). As expected the normalized torque reduces over the optimization by 42% (compare section S5). The reduction in normalized torque as an efficiency measure confirms our hypothesis, that matching the control task dynamics to the system's natural dynamics has beneficial effects on energy requirements.



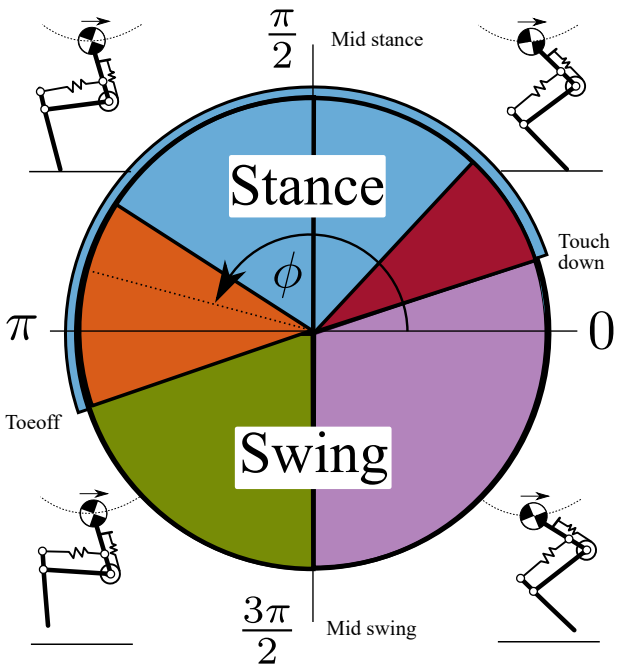
(a)



(b)



(c)



(d)

Fig. 2: **Schematic depiction of the neuroelasticity and neuroplasticity framework. a,** Schematic depiction of short-term neuroelasticity and long-term neuroplasticity. Neuroelasticity (green) mitigates stochastic short term perturbations (red) like a pot hole that disturb the system (spring) from its desired state (dashed line). Neuroelastic activity is reversible and only active when a perturbation is present. Just like a spring only deflects as long as an external force is active and then returns to its initial state. Neuroplasticity (yellow) changes the system behavior permanently to adapt to long-term active stimuli from the environment. If the same perturbation is frequently present, the system adapts to the perturbation. In our example the spring adapts its set point (spring length) and stiffness (spring thickness). This way an initial desired system state that might be encoded in initial control design, can be adapted to better deal with perturbations throughout its life span as well as to changing environments. After the neuroplastic adaptation the spring now deflects less (bottom right green). In this study we measure the amount of neuroelastic activity during level walking of a quadruped robot. We quantify the mismatch between the robot's natural dynamics and the control task dynamics based on the amount of active feedback. By minimizing the neuroelastic activity through optimization we neuroplastically change the control dynamics to increase dynamics matching. **b,** Control structure of quadruped robot Morti. **c,** Flowchart of the matching approach. The neuroelastic activity mitigates short term perturbations through sparse contact feedback from the FootTile contact sensors. We measure the amount of neuroelastic activity as a proxy for the mismatching of dynamics. Over a longer time window the optimizer minimizes the neuroelastic activity to neuroplastically match the control task dynamics of the CPG to the robot's natural dynamics. **d,** Diagram of a step cycle in phase space. Colored sections for feedback mechanisms: late touchdown (red) later than the desired touchdown time TD ($\delta_{\text{overSwing}}$), late toe-off (yellow) later than the desired toeoff time TO ($\delta_{\phi,\text{knee}}$), early toe-off (green), early touchdown (purple). Stance phase from touchdown to toeoff is shaded blue.

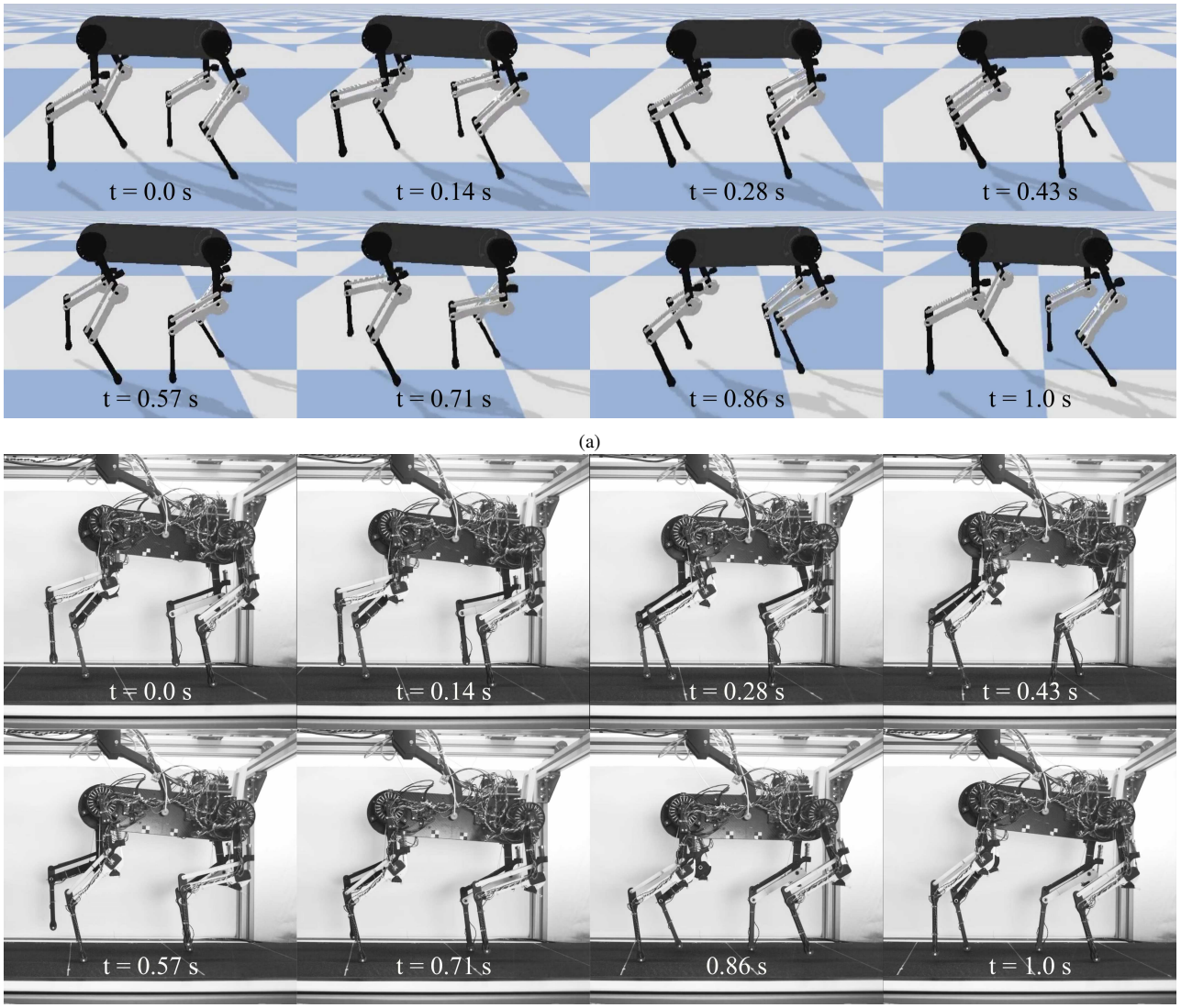


Fig. 3: **Snapshots of simulation and hardware rollout. a,** Snapshots of the simulated quadruped robot Morti walking during one of the optimization rollouts in the PyBullet multibody simulation. **b,** Snapshots of quadruped robot Morti walking on a treadmill during one of the hardware rollouts.

II. Discussion

In this paper, we develop an approach to measure and improve the matching between control and natural dynamics. We measure the neuroelastic activity of feedback to correct for mismatched timing during locomotion on even ground where no external perturbations should be present. We then neuroplastically adapt the control dynamics to match Morti's natural dynamics through Bayesian optimization in simulation. The optimization results are then validated in hardware experiments. We show that sparse feedback from a simple contact sensor can be used as a proxy to measure the dynamics mismatching. Already 150 optimization rollouts suffice to reach good locomotion performance with an optimization duration of roughly one hour. Our reported rollout numbers are orders of magnitude lower compared to results reported for machine learning approaches [20], [21]. We also find that matching dynamics is beneficial for energy efficiency. We calculate a normalized performance measure which shows a decrease in power requirements.

The designed passive behavior our robot Morti enable a simple matched CPG control structure to leverage the natural dynamics of the leg design. Through sparse, binary feedback from touch sensors the controller is able to neuroelastically mitigate the perturbations stemming from initial mismatching. Through the interplay of natural dynamics and the matched CPG, Morti can achieve convincing locomotion performance on inexpensive hardware with lower computation power and with lower control and sensor bandwidth compared to state of the art model-based locomotion controllers.

Examining the reward (Figure 5b) shows that through dynamics matching and the minimization of neuroelastic activity (J_{feedback}) the robot is able to travel longer distances in the given time as shown by the improved distance reward (J_{distance}). The body pitch reward (J_{pitch}) does not minimize

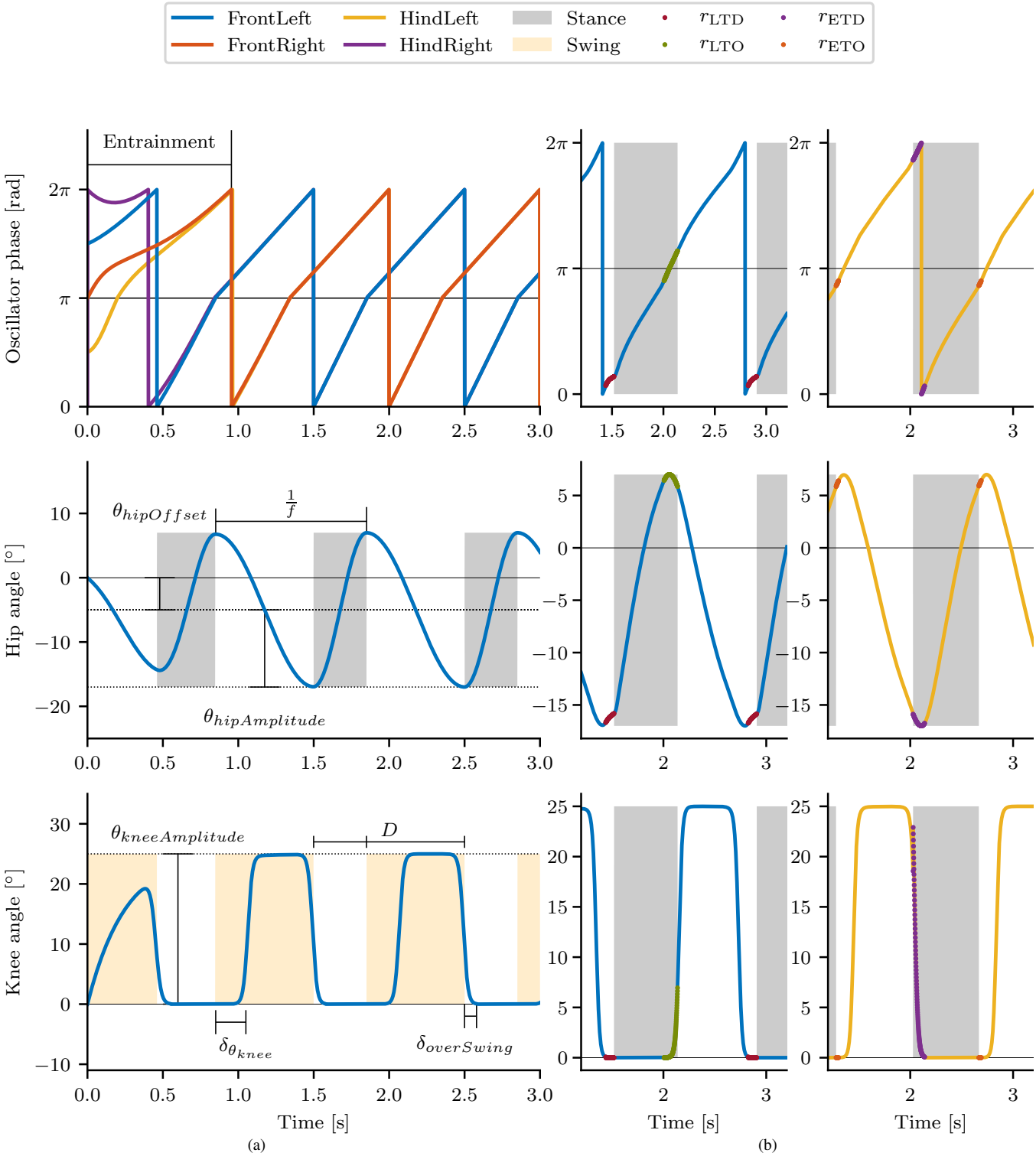


Fig. 4: **CPG parameters and neuroelastic activity. a,** Example CPG output for four coupled oscillators and the generated trajectories. Top are the coupled phases, middle and bottom are the hip and knee joint trajectories for one oscillator with their respective CPG parameters p_m (Table S2). Parameters here are $D = 0.35$, $\delta_{\phi,\mathrm{knee}} = 0.3$, $\delta_{\mathrm{overSwing}} = 0.2$, $f = 1$. **b,** Simulation results showing the four feedback mechanisms (same color coding as Figure S1). Data is shown for the front left (blue) and front right (orange) leg. Late touchdown (red) on the front left leg phase shows the phase delay to wait for touchdown. Early touchdown (purple) on the right leg shows the knee pull up reflex. Late toeoff (yellow) is shown on he left leg. Early toeoff (green) is shown on the right leg. Stance phase is shaded gray.

much over the optimization. This is expected because the CPG cannot actively control body pitch. The reward terms for periodicity ($J_{\mathrm{periodicity}}$) and contact (J_{contact}) are used as penalty terms for unnatural gaits. They are an order of magnitude lower compared to the remaining reward terms and only peak for less performant rollouts.

In the normalized torque measure, the robot benefits from the increase in distance reward as well as a reduction in required torque through the matching of control task and natural dynamics. Even though the robot increases its speed more than twofold, the required torque does not increase, instead, the normalized torque reduces with a trend comparable to the reward term because the improved dynamics matching enables the controller to leverage the natural dynamics to achieve better performance (Figure 6).

While other studies report problems with transferring simulation results to hardware (sim2real gap) [56]–[58] we could successfully transfer our simulation results to the Morti hardware without post-transfer modifications. The hardware performance is comparable both quantitatively and in qualitative observation of the resultant gaits (supplementary videos). We believe that, because the joint torques in our robot are not calculated from, possibly inaccurate, model parameters, the sim2real gap is not as visible here. Learned controllers that directly influence joint torques and leg forces might suffer more from the sim2real gap because smaller inaccuracies in between model and hardware behavior can have a direct effect on the forces exerted onto the robot. However, more research will be required to understand the transferability of results for underactuated robots with strong natural dynamics.

In future work, we consider reformulating the CPG, taking body pitch into account when generating the hip trajectories [36]. With an inertial measurement unit (IMU) the body pitch could be fed back into the CPG. In the current formulation, the CPG assumes no body pitch and relies on the robustness the passive elasticity adds to the system to compensate the existing body pitch. Additionally, abduction/adduction degrees of freedom could be added to the robot to enable 3D locomotion without a guiding mechanism. The optimization loop could be implemented to run online on the robot's computer. With online optimization and 3D locomotion, it would become possible to investigate the adaptation of the CPG dynamics to changing ground conditions and surface properties over extended time windows.

Many contemporary legged robot designs avoid engineering strong natural dynamics. They are instead used to investigate model-based control approaches where passive dynamics are hindering. These robots perform well as long as good knowledge about the environment and the robot state are available. Yet so far, legged robots perform poorly in natural environments where locomotion conditions are complex, and knowledge about substrate and terrain is sparse. Legged robots that leverage mechanics to create favorable dynamics ('computational morphology' [72]) require low control effort [59], and are less dependent on sensory feedback. Our framework provides a blueprint of how legged robots with strong engineered natural dynamics can successfully adapt their control dynamics to the robot's natural dynamics with sparse feedback.

III. Methods

For both experimentation and simulation, we design and implement quadruped robot Morti. Morti has a monoarticular knee spring and a biarticular spring between hip and foot that provides series elastic behavior [8]. The robot is controlled by a closed-loop CPG. Through reflex-like feedback mechanisms, the robot can neuroelastically mitigate short-term perturbations. To minimize the neuroelastic activity we implement a Bayesian optimizer that neuroplastically matches the control task dynamics to the robot's natural dynamics.

A. Robot Mechanics

The robot consists of four 'biarticular legs' [8, Fig. 1B] mounted to a carbon-fiber body. Each leg has three segments: femur, shank and foot segment. Femur and foot segment are connected through a spring-loaded parallel mechanism mimicking the biarticular muscle-tendon structure formed by the gastrocnemius muscle-tendon group in quadruped animals [60]. A knee spring inspired by the patellar tendon in animals provides passive elasticity of the knee joint.

The robot walks on a treadmill and is constrained to the sagittal plane by a linear rail that allows body pitch (Figure 1b). The robot is instrumented with joint angle sensors, position sensors and the treadmill speed sensor. To measure ground contact, four FootTile sensors [61] are mounted on the robot's feet. Through a threshold, these analog pressure sensors can be used to determine if the robot established ground contact. Detailed descriptions of the experimental setup can be found in section S1.

B. Simulation

We implement the simulation in PyBullet [56], a multibody simulator based on the bullet physics engine (Figure 3a). The robot mechanics are derived from the mechanical robot and its CAD model (Table S1). To increase the match between robot hardware and simulation, we implement motor limits and the motor controller to resemble the real actuator limits [20]. The simulation runs at 1 kHz, the CPG control loop is running at 500 Hz and ground contacts are polled at 250 Hz to resemble the hardware implementation.

C. CPG

The CPG used in this work is a modified Hopf oscillator [34] that is modeled in phase space. Similar to its biological counterpart, it can be entrained through feedback from external sensory input or from internal coupling to neighboring nodes. Based on the desired phase shifts in between oscillator nodes a variety of gaits can be implemented by adapting the phase difference matrix while keeping the network dynamics identical (section S6). The joint trajectories generated by the CPG are described by eight parameters (Figure 4a). Hip offset ($\Theta_{\mathrm{hipOffset}}$) and hip amplitude ($\Theta_{\mathrm{hipAmplitude}}$) describe the

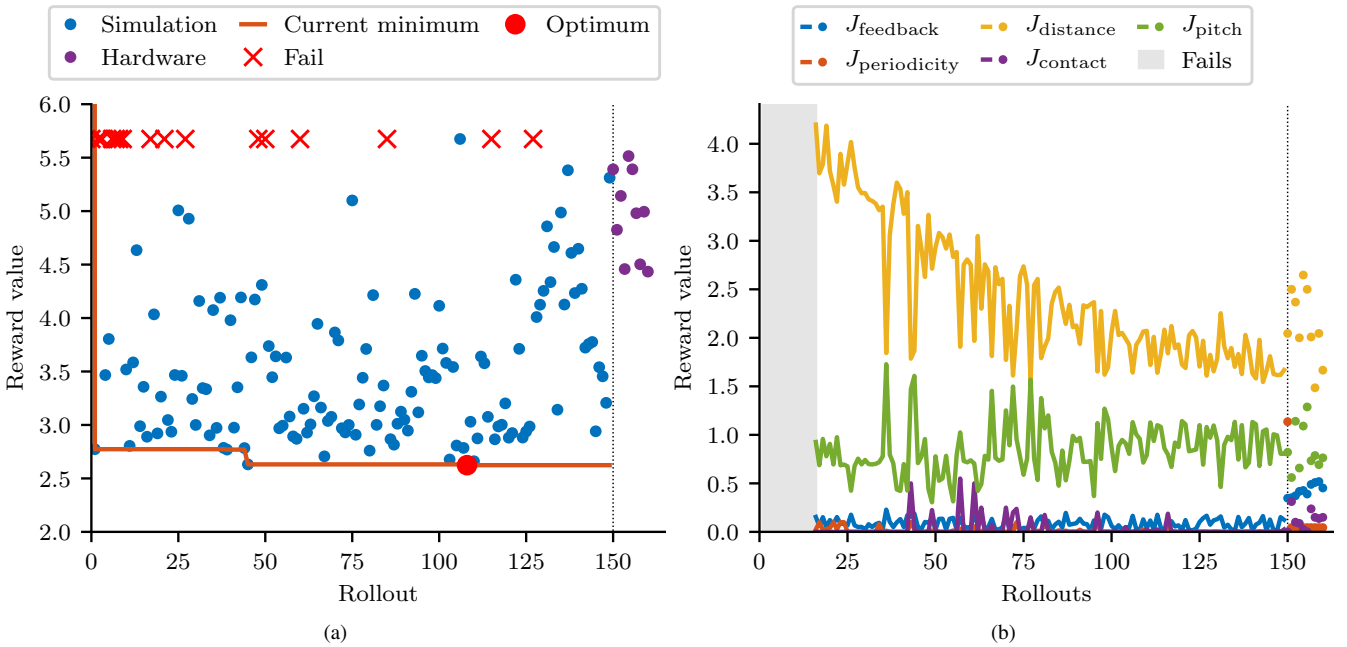


Fig. 5: **Results from neuroplastic adaptation. a,** Reward function for the Bayesian optimization. Because Bayesian optimization is not monotonically minimizing the reward function, the current minimum is shown in orange. The global minimum is shown as a red dot, failures as crosses. **b,** Individual reward values from the different reward function terms (Table S3). The individual reward terms show similar results between the simulation results (lines) and the hardware samples (dots). The mean hardware reward values are similar to the optimal rewards from simulation.

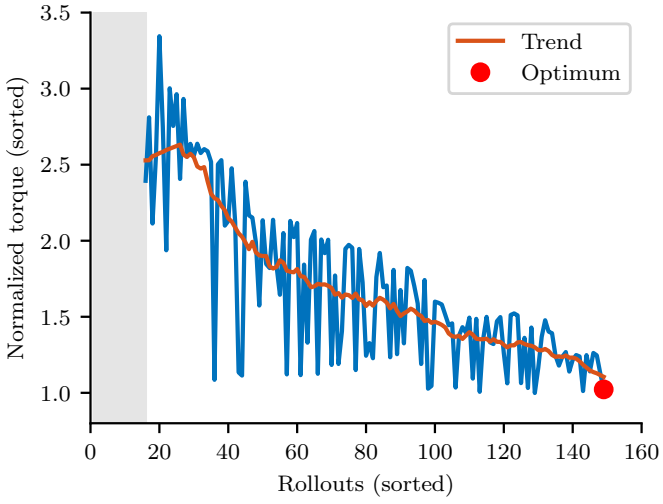


Fig. 6: **Normalized torque performance measure.** Normalized Torque ($\frac{\text{torque}}{\text{speed}}$) reduces over optimization. By matching the control task dynamics to the robot's natural dynamics the required energy for locomotion reduces. Because the controller learns to exploit the robot's natural dynamics, less energy is used to achieve the desired behavior. The normalized torque was not part of the cost function but minimizes because the matching increases (compare section S5).

hip trajectory. Knee amplitude ($\Theta_{\text{kneeOffset}}$) describes the knee flexion. The frequency f determines the robot's overall speed, duty factors (D) describe the ratio of stance phase to flight phase, knee phase shift ($\delta_{\phi,\text{knee}}$) describes the phase shift between hip protraction and knee flexion and overswing ($\delta_{\text{overSwing}}$) describes the amount of swing leg retraction [62]. The mathematical description for the CPG dynamics can be found in section S2.

D. *Neuroelasticity*

As the CPG implemented here is written as a model-free feed-forward network it can be difficult to find parameters that lead to viable gaits with given robot dynamics. Essentially the CPG commands desired trajectories without knowledge of the robot's natural dynamics. In the worst case the CPG could command behavior the robot cannot fulfill because of its own natural dynamics and mechanical limitations like inertia, motor speed, or torque limitations. To fix this shortcoming, feedback can be used to mitigate the differences between desired and measured behavior. One possible feedback that has been shown to aid in entrainment and can mitigate perturbations is foot contact information [32]. This contact information can be integrated into the CPG to measure timing differences between the desired and measured trajectories.

The trajectories created by the CPG can be influenced through feedback either by changing the CPG dynamics, meaning accelerating or decelerating the CPG's phases. Alternatively, feedback can influence the generated joint angle trajectories. During a step cycle (Figure 2d), contact signals can be used for several feedback mechanisms (Figure S1). The feedback mechanisms react to timing discrepancies for the touchdown and toeoff events and correct the CPG trajectories if the robot establishes or loses ground contact earlier or later than instructed by the CPG. In this work, we implement a phase deceleration for delayed touchdown, early knee flexion when ground contact is lost too early, a phase deceleration

when knee flexion is delayed, and a knee pull-up reflex in combination with disabling the hip motor when a leg hits the ground too early. An in-depth description of the feedback mechanisms can be found in section S3.

E. Neuroplasticity

To match the CPG to the robot dynamics we want to tune the CPG parameters p_m to achieve optimal performance. To do so we evaluate the performance of the robot for a number of steps. The time scale of the optimization is designed to be much bigger compared to the frequency of the neuroelastic activity mechanisms ($\leq$0.1 Hz vs. $\geq$100 Hz). Consequently the effects of the neuroelastic activity are minimized and small perturbations within one step are not captured in the neuroplastic optimization that will only improve long term performance.

To achieve long-term (close to) optimal behavior we use Bayesian optimization for its global optimization capabilities, data efficiency and robustness to noise [53], [54].

1) Bayesian optimization: Bayesian optimization is a black box optimization approach that uses Gaussian kernels for function approximation. It is model-free, derivative-free and has been used successfully in many robotic optimization approaches [63]–[65]. Bayesian optimization is favored over other data-driven optimization and learning approaches because of its data efficiency for ≤ 10 parameters. While in simulation data efficiency is not a big issue, given enough computation, hardware experiments are expensive in terms of wear, risk and time.

We implement a Bayesian optimizer based on skopt gp_minimize [66]. The optimizer evaluates the PyBullet simulation for 10 s ($\approx$ 10 step cycles) of each rollout with a reward function. The CPG has 10 s to entrain itself from its initial condition (standing still) and to get the robot to run before the evaluation period begins. One complete rollout therefor takes 20 s. We optimize for 15 rollouts with random CPG parameters before approximating the reward function. Then we optimize for 135 rollouts with "gp_hedge" acquisition function which is a probabilistic choice of lower confidence bound (*LCB*), negative expected improvement (*EI*) and negative probability of improvement (*PI*).

To reduce complexity we limit the parameter space to six parameters. The six parameters are $\Theta_{\mathrm{hipOffset}}$, $\Theta_{\mathrm{hipAmplitude}}$, D_{front} and D_{hind}, $\delta_{\phi,\mathrm{knee}}$ and $\delta_{\mathrm{overSwing}}$. More parameters would likely increase performance more, but will also lead to more corner cases where the selected cost function can be exploited by the optimizer and result in unnatural gaits like skipping gaits or gaits where the feet drag on the ground. For this proof of concept, we chose independent duty factors D_{front} and D_{hind} to allow some front-hind asymmetry that can help the optimizer find gaits that reduce body pitch. Where only one CPG parameter is selected, the parameter is used for all legs. For simplicity, we also fix the frequency to $f = 1$ Hz to reduce experimental cost in terms of hardware wear from violent motions at high speed. The hip amplitude $\Theta_{\mathrm{kneeAmplitude}}$ is set to 30° to ensure adequate ground clearance.

2) Cost function: We evaluate the robot based on a cost function comprised of three major components. The first component influences the matching behavior (J_{feedback}), specifically the amount of neuroelastic activity that the robot uses during a rollout. The second component measures effective forward locomotion (J_{distance}) to provide meaningful results. The third component ensures a gait comparable to the gaits observed in animals and serves mostly as a penalty for 'unnatural gait characteristics'. It enforces, that the robot moves with little body pitch (J_{pitch}), enforces only one contact phase per leg and step (J_{contact}) to prevent dragging and skipping and only takes one step per stride and leg ($J_{\mathrm{periodicity}}$). Further description can be found in section S4.

F. Hardware rollouts

To validate the optimal set of CPG parameters p_m from simulation, we test the same parameters on the hardware robot. The hardware controller has the same neuroelastic mechanisms that were previously described in subsection III-D. We test 10 parameter sets and randomly vary the CPG parameters obtained from simulation by $\leq 10\%$ to validate the hardware reward function around the optimal point found in simulation. We then evaluate the robot performance with the same reward function used for the simulation. Like the simulation, the robot CPG is entrained in air and the performance is only measured for 10 s after the robot converged to a stable gait. Videos of the robot walking can be found in the supplementary material.

IV. Acknowledgment

We thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting the academic development of Felix Ruppert. This work was made possible thanks to a Max Planck Group Leader grant awarded to ABS by the Max Planck Society. The authors thank ZWE robotics for support with 3D printing, the MPI machine shop for support with metal manufacturing, Majid Khadiv and Ludovic Righetti for the discussions concerning Bayesian optimization, Alborz Sarvestani for the many fruitful discussions and Robin Petereit for his input concerning Morti's firmware.

V. Contribution

FR contributed to concept, design, experiments, data analysis and writing. ABS contributed to concept, feedback and supervision.

VI. Competing Interests

The authors declare no competing interests.

VII. Data Availability

All data will be publicly available upon request.

VIII. Code Availability

All relevant code is available for review and will be made available publicly upon publication. The robot design files will also be made publicly available upon publication.

References

[1] R. Alexander, "Elastic energy stores in running vertebrates," *American Zoologist*, vol. 24, no. 1, pp. 85–94, Feb. 1984.

[2] R. Blickhan, "The spring-mass model for running and hopping," *Journal of biomechanics*, vol. 22, no. 11-12, pp. 1217–1227, 1989.

[3] E. Westervelt, J. Grizzle, and D. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, Jan. 2003.

[4] M. A. Daley, "Running over rough terrain: Guinea fowl maintain dynamic stability despite a large unexpected change in substrate height," *Journal of Experimental Biology*, vol. 209, no. 1, pp. 171–187, Jan. 2006.

[5] D. Renjewski, A. Spröwitz, A. Peekema, *et al.*, "Exciting engineered passive dynamics in a bipedal robot," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1244–1251, Oct. 2015.

[6] X. Luo and W. Xu, "Planning and control for passive dynamics based walking of 3d biped robots," *Journal of Bionic Engineering*, vol. 9, no. 2, pp. 143–155, Jun. 2012.

[7] A. Ruina, *Passive dynamics is a good basis for robot design and control, not!* 2017.

[8] F. Ruppert and A. Badri-Spröwitz, "Series elastic behavior of biarticular muscle-tendon structure in a robotic leg," *Frontiers in Neurorobotics*, vol. 13, Aug. 2019.

[9] A. Spröwitz, A. Tuleu, M. Vespignani, *et al.*, "Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 932–950, Jun. 2013.

[10] B. Katz, J. D. Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, May 2019.

[11] H. Lee and N. Hogan, "Time-varying ankle mechanical impedance during human locomotion," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 5, pp. 755–764, Sep. 2015.

[12] R. Tedrake, T. Weirui, Z. H. Sebastian, *et al.*, "Learning to walk in 20 minutes," 224.

[13] P. A. Bhounsule, J. Cortell, A. Grewal, *et al.*, "Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge," *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1305–1321, Jun. 2014.

[14] H. Geyer, A. Seyfarth, and R. Blickhan, "Spring-mass running: Simple approximate solution and application to gait stability," *Journal of Theoretical Biology*, vol. 232, no. 3, pp. 315–328, Feb. 2005.

[15] ——, "Compliant leg behaviour explains basic dynamics of walking and running," *Proceedings of the Royal Society B: Biological Sciences*, vol. 273, no. 1603, pp. 2861–2867, Nov. 2006.

[16] J. Rummel and A. Seyfarth, "Stable running with segmented legs," *The International Journal of Robotics Research*, vol. 27, no. 8, pp. 919–934, Aug. 2008.

[17] S. A. Migliore, E. A. Brown, and S. P. DeWeerth, "Novel nonlinear elastic actuators for passively controlling robotic joint compliance," *Journal of Mechanical Design*, vol. 129, no. 4, pp. 406–412, Apr. 2006.

[18] M. H. Raibert, "Running with symmetry," in *Autonomous Robot Vehicles*, Springer New York, 1986, pp. 45–61.

[19] H.-W. Park and S. Kim, "The MIT cheetah, an electrically-powered quadrupedal robot for high-speed running," *Journal of the Robotics Society of Japan*, vol. 32, no. 4, pp. 323–328, 2014.

[20] J. Tan, T. Zhang, E. Coumans, *et al.*, "Sim-to-real: Learning agile locomotion for quadruped robots," in *Robotics: Science and Systems XIV*, Robotics: Science and Systems Foundation, Jun. 2018.

[21] J. Siekmann, S. Valluri, J. Dao, *et al.*, "Learning memory-based control for human-scale bipedal locomotion," Jun. 3, 2020. arXiv: 2006.02402 [cs.RO].

[22] C. Hubicki, J. Grimes, M. Jones, *et al.*, "ATRIAS: Design and validation of a tether-free 3d-capable spring-mass bipedal robot," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1497–1521, Jul. 2016.

[23] J. Rummel, Y. Blum, and A. Seyfarth, "Robust and efficient walking with spring-like legs," *Bioinspiration & Biomimetics*, vol. 5, no. 4, p. 046 004, Nov. 2010.

[24] G. Kenneally, A. De, and D. E. Koditschek, "Design principles for a family of direct-drive legged robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 900–907, Jul. 2016.

[25] A. Bicanski, D. Ryczko, J. Knuesel, *et al.*, "Decoding the mechanisms of gait generation in salamanders by combining neurobiology, modeling and robotics," *Biological Cybernetics*, vol. 107, no. 5, pp. 545–564, Feb. 2013.

[26] N. Dominici, Y. P. Ivanenko, G. Cappellini, *et al.*, "Locomotor Primitives in Newborn Babies and Their Development," *Science*, vol. 334, no. 6058, pp. 997–999, Nov. 2011, ISSN: 0036-8075, 1095-9203.

[27] R. Grasso, Y. P. Ivanenko, M. Zago, *et al.*, "Distributed plasticity of locomotor pattern generators in spinal cord injured patients," *Brain*, vol. 127, no. 5, pp. 1019–1034, May 2004.

[28] E. Marder and D. Bucher, "Central pattern generators and the control of rhythmic movements," *Current Biology*, vol. 11, no. 23, R986–R996, Nov. 2001.

[29] K. Matsuoka, "Mechanisms of frequency and pattern control in the neural rhythm generators," *Biological Cybernetics*, vol. 56, no. 5-6, pp. 345–353, Jul. 1987.

[30] E. Bizzi, M. C. Tresch, P. Saltiel, *et al.*, "New perspectives on spinal motor systems," *Nature Reviews Neuroscience*, vol. 1, no. 2, pp. 101–108, Nov. 2000.

[31] M. H. Dickinson, "How animals move: An integrative view," *Science*, vol. 288, no. 5463, pp. 100–106, Apr. 2000.

[32] L. Righetti and A. J. Ijspeert, "Pattern generators with sensory feedback for the control of quadruped locomotion," in *2008 IEEE International Conference on Robotics and Automation*, IEEE, May 2008.

[33] F. Xie, Y. Zhong, R. Du, *et al.*, "Central pattern generator (CPG) control of a biomimetic robot fish for multimodal swimming," *Journal of Bionic Engineering*, vol. 16, no. 2, pp. 222–234, Mar. 2019.

[34] A. J. Ijspeert, A. Crespi, D. Ryczko, *et al.*, "From swimming to walking with a salamander robot driven by a spinal cord model," *Science*, vol. 315, no. 5817, pp. 1416–1420, Mar. 2007.

[35] R. Thandiackal, K. Melo, L. Paez, *et al.*, "Emergence of robust self-organized undulatory swimming based on local hydrodynamic force sensing," *Science Robotics*, vol. 6, no. 57, eabf6354, Aug. 2021.

[36] G. Sartoretti, S. Shaw, K. Lam, *et al.*, "Central pattern generator with inertial feedback for stable locomotion and climbing in unstructured terrain," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2018.

[37] P. Mitteroecker and E. Stansfield, "A model of developmental canalization, applied to human cranial form," *PLOS Computational Biology*, vol. 17, no. 2, S. L. K. Pond, Ed., e1008381, Feb. 2021.

[38] G. Bledt, M. J. Powell, B. Katz, *et al.*, "MIT cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2018.

[39] Y. Gong, R. Hartley, X. Da, *et al.*, "Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway," in *2019 American Control Conference (ACC)*, IEEE, Jul. 2019.

[40] F. Grimminger, A. Meduri, M. Khadiv, *et al.*, "An open torque-controlled modular robot architecture for legged locomotion research," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3650–3657, Apr. 2020.

[41] M. Oliveira, V. Matos, C. P. Santos, *et al.*, "Multi-objective parameter cpg optimization for gait generation of a biped robot," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3130–3135.

[42] R. Calandra, A. Seyfarth, J. Peters, *et al.*, "An experimental comparison of bayesian optimization for bipedal locomotion," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1951–1958.

[43] M. H. Yeganegi, M. Khadiv, S. A. A. Moosavian, *et al.*, "Robust humanoid locomotion using trajectory optimization and sample-efficient learning," in *Proceedings International Conference on Humanoid Robots*, IEEE, Oct. 2019.

[44] J. Bongard, V. Zykov, and H. Lipson, "Resilient machines through continuous self-modeling," *Science*, vol. 314, no. 5802, pp. 1118–1121, Nov. 2006.

[45] Y. Fukuoka, H. Kimura, Y. Hada, *et al.*, "Adaptive dynamic walking of a quadruped robot 'tekken' on irregular terrain using a neural system model," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, IEEE.

[46] J. Buchli and A. J. Ijspeert, "Self-organized adaptive legged locomotion in a compliant quadruped robot," *Autonomous Robots*, vol. 25, no. 4, pp. 331–347, Jul. 2008.

[47] J. Buchli, L. Righetti, and A. J. Ijspeert, "A dynamical systems approach to learning: A frequency-adaptive hopper robot," in *Advances in Artificial Life*, Springer Berlin Heidelberg, 2005, pp. 210–220.

[48] B. A. Pearlmutter, "Learning state space trajectories in recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 263–269, Jun. 1989.

[49] S. Heim, F. Ruppert, A. A. Sarvestani, *et al.*, "Shaping in practice: Training wheels to learn fast hopping directly in hardware," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2018.

[50] T. Matsubara, J. Morimoto, J. Nakanishi, *et al.*, "Learning cpg-based biped locomotion with a policy gradient method," in *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, 2005, pp. 208–213.

[51] Y. Nakamura, T. Mori, and S. Ishii, "Natural policy gradient reinforcement learning for a cpg control of a biped robot," in *Parallel Problem Solving from Nature - PPSN VIII*, X. Yao, E. K. Burke, J. A. Lozano, *et al.*, Eds., Springer Berlin Heidelberg, 2004.

[52] X. B. Peng, E. Coumans, T. Zhang, *et al.*, "Learning agile robotic locomotion skills by imitating animals," in *Robotics: Science and Systems XVI*, Robotics: Science and Systems Foundation, Jul. 2020.

[53] R. Calandra, A. Seyfarth, J. Peters, *et al.*, "Bayesian optimization for learning gaits under uncertainty," *Annals of Mathematics and Artificial Intelligence*, vol. 76, no. 1-2, pp. 5–23, Jun. 2015.

[54] J. Mockus, *Bayesian Approach to Global Optimization*. Springer Netherlands, Dec. 6, 2012, 270 pp., ISBN: 9789400909090.

[55] D. Owaki and A. Ishiguro, "A quadruped robot exhibiting spontaneous gait transitions from walking to trotting to galloping," *Scientific Reports*, vol. 7, no. 1, Mar. 2017.

[56] E. Coumans and Y. Bai, *Pybullet, a python module for physics simulation for games, robotics and machine learning*, `http://pybullet.org`, 2016–2019.

[57] K. Rosser, J. Kok, J. Chahl, *et al.*, "Sim2real gap is non-monotonic with robot complexity for morphology-

in-the-loop flapping wing design," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2020.

[58] E. Heiden, D. Millard, E. Coumans, *et al.*, "NeuralSim: Augmenting differentiable simulators with neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[59] D. F. B. Haeufle, M. Günther, G. Wunner, *et al.*, "Quantifying control effort of biological and technical movements: An information-entropy-based approach," *Physical Review E*, vol. 89, no. 1, Jan. 2014.

[60] H. Witte, R. Hackert, K. E. Lilje, *et al.*, "Transfer of biological principles into the construction of quadruped walking machines," in *Proceedings of the Second International Workshop on Robot Motion and Control. RoMoCo'01 (IEEE Cat. No.01EX535)*, Poznan Univ. Technol, 2001.

[61] F. Ruppert and A. Badri-Spröwitz, "FootTile: A rugged foot sensor for force and center of pressure sensing in soft terrain," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2020.

[62] A. Seyfarth, H. Geyer, and H. Herr, "Swing-leg retraction: A simple control model for stable running," *J Exp Biol*, vol. 206, no. 15, pp. 2547–2555, Aug. 2003.

[63] M. Gianni, M. A. R. Garcia, and F. Pirri, "Learning the dynamics of articulated tracked vehicles," en, 2016.

[64] A. Marco, P. Hennig, J. Bohg, *et al.*, "Automatic LQR tuning based on gaussian process global optimization," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2016.

[65] T. Seyde, J. Carius, R. Grandia, *et al.*, "Locomotion planning through a hybrid bayesian trajectory optimization," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, May 2019.

[66] T. Head, M. Kumar, H. Nahrstaedt, *et al.*, *Scikit-optimize/scikit-optimize*, 2020.

S1. Robot Mechanics

The robot consists of four 'biarticular legs' [8, Fig. 1B] mounted to a carbon fiber compound plate (CFK sandwich, *carbon-vertrieb*). The body structure provides high bending and torsion stiffness at low weight. Each leg is articulated by two brushless outrunner motors (MN7005, *tmotors*). The hip motor is geared through a 1:5 planetary gearbox (RS3505S, *Matex*), the knee motor is geared through a 1:12 one-directional cable drive mechanism to flex the knee joint. The knee extends through the knee spring that tensions during flexion. The knee and ankle joints of the robot are instrumented with rotary encoders (AEAT8800, *Broadcom*). A hall-effect switch (DRV5023, *Texas Instruments*) between the body and femur segment provides a reference to initiate the angle measurements of the hip joints. Hard stops at the robot's knee and ankle joints prevent overextension during stance phase. Four hall-effect current sensors (ACS723, *Alegro Microsystems*) measure input currents to the motor driver boards. The robot is controlled by a Raspberry Pi 4 through a custom-made shield. The shield consists of a SPI GPIO expander (MCP23017, *Microchip* and four tri-state buffers (74HC125, *nexperia*). The shield connects four SPI controlled custom brushless motor drivers [40], the hall switches and the joint encoders to the computer. Two external 12 V 80 A lead batteries supply motors and the computer with power.

Here we focus on motion in the sagittal plane. The robot is therefor constrained to motion in the sagittal plane by a linear rail and lever mechanism that also allows body pitch around the robot's center of mass (COM). The robot walks on an off-the-shelf recreational treadmill (TM500S, *Christopeit*) that is retrofitted with a motor controller (DPCANIE, *amc*). Treadmill speed is measured with a rotary encoder (AEAT8800, *Broadcom*). The robot is connected to the treadmill with a linear rail (SSEB, *Misumi*) and a lever mechanism. The linear rail is equipped with a linear encoder (AS5311, *ams*). The experimental setup can be seen in Figure 1b.

To measure ground contact each foot is equipped with a FootTile sensor [61] connected to a I^2C multiplexer (TCA9548A, *Texas Instruments*). The sensor dome is a half-cylinder with a half-cylinder air cavity that houses the sensor. We do not expect forces in the lateral direction due to the guiding mechanism. Therefor the sensor domes are laterally symmetric (Figure 1b). As the FootTile sensors are analog pressure sensors, we define a threshold to measure when the sensors are in contact with the ground. The sensors read values between 98.5 and 99 kPa when not in contact with the ground. We define 100 kPa as the contact threshold.

All data on the robot is sampled at 500 Hz except for the FootTile data is sampled at 250 Hz because of limitations in the pressure sensor hardware.

TABLE S1: **Robot parameters**

Parameter	Value
Quadruped robot Morti	
Leg length	408 mm
Body length	350 mm (shoulder - hip)
Width	120 mm (shoulder - shoulder)
Mass	3.6 kg
Hip gear ratio	1:5
Knee gear ratio	1:12
Max. hip torque	6.2 Nm
Max. knee torque	14.9 Nm
Knee cam radius	30 mm
Knee spring stiffness	10.89 $\frac{\text{N}}{\text{mm}}$
Biarticular spring stiffness	9.8 $\frac{\text{N}}{\text{mm}}$
FootTile Sensor	
Dome material	Vytaflex 40
Dome diameter	10 mm
Dome width	10 mm
Sample rate	250 Hz
Resolution	12 bit

S2. CPG

The CPG is described by a system of coupled differential equations:

$$\dot{\phi_j} = \Omega_j \tag{S1}$$

for phase vector

$$\phi = \begin{pmatrix} \phi_{\text{frontLeft}} \\ \phi_{\text{frontRight}} \\ \phi_{\text{hindLeft}} \\ \phi_{\text{hindRight}} \end{pmatrix}$$

where ϕ is the oscillator phase vector and Ω is the angular velocity vector.

$$\dot{\phi_j} = 2\pi f + \sum_{k=1}^{N} \alpha_{\text{dyn},j,k} \cdot \mathbf{C}_{jk} \cdot \sin(\phi_k - \phi_j - \mathbf{\Phi}_{jk}) \tag{S2}$$

where f is the frequency, $\alpha_{\text{dyn},j,k}$ is the conversion constant of the network dynamics between nodes j and k, $\mathbf{C}_{jk}$ is the coupling matrix weight between nodes j and k, $\mathbf{\Phi}_{jk}$ is the desired phase difference matrix value between nodes j and k.

$$\phi_j = \begin{cases} \frac{\phi_j}{2 \cdot D} & \phi_j < 2\pi \cdot D \\ \frac{\phi_j + 2\pi(1-2\cdot D)}{2(1-D)} & else \end{cases} \tag{S3}$$

$$D = \frac{t_{\text{stance}}}{(t_{\text{stance}} + t_{\text{flight}})} \tag{S4}$$

where ϕ_j is the j[th] end-effector phase, D is the duty factor, t_{stance} is the duration of stance phase and t_{flight} is the duration of flight phase.

$$\Theta_{\text{hip,des,j}} = \Theta_{\text{hipOffset},j} + \Theta_{\text{hipAmplitude},j} \cdot \cos(\phi_j) \tag{S5}$$

Θ_{hip} is the hip reference trajectory vector, $\Theta_{\text{offset},j}$ is the hip offset and $\Theta_{\text{hipAmplitude},j}$ is the hip amplitude of node j. Depending on the gait symmetry, Θ_{offset} and $\Theta_{\text{hipAmplitude}}$ are also only equal in legs that share the same gait symmetry. For example, in trot, all four legs move symmetrically so all hip amplitudes ($\Theta_{\text{hipAmplitude},j}$) are equal, where as in bound only the front legs and the hind legs move symmetrically, also $\Theta_{\text{hipAmplitude},1}$ are only equal in the front and hind.

TABLE S2: **CPG parameters as used in Equation S1 - Equation S12.**

Parameter p_m	Symbol	Description
p_0	$\Theta_{\text{hipOffset}}$	Hip offset
p_1	$\Theta_{\text{hipAmplitude}}$	Hip amplitude
p_2	$\Theta_{\text{kneeAmplitude}}$	Knee amplitude
p_3	$\Theta_{\text{kneeOffset}}$	Knee offset
p_4	f	Frequency
p_5	$\delta_{\phi,\text{knee}}$	Knee phase shift
p_6	D	Duty factor
p_7	$\delta_{\text{overSwing}}$	Knee overswing

$$\Theta_{\text{knee,des,j}} = \left(\frac{1}{1 + e^{\frac{\phi_j - \phi_{\text{flex}}}{T_{\text{flex}}}}} - \frac{1}{1 + e^{\frac{\phi_j - \phi_{\text{ext}}}{T_{\text{ext}}}}} \right) \cdot \Theta_{\text{kneeAmplitude}} + \Theta_{\text{kneeOffset}} \tag{S6}$$

$\Theta_{\text{knee,des}}$ is the knee reference trajectory vector, ϕ_{ext} and ϕ_{flex} are phase shifts for the onset of knee flexion and extension and T_{flex} and T_{ext} are the time constant of the transient flexion and extension behavior.

$$\phi_{\text{kneeShift}} = 2\pi \cdot D + \delta_{\phi,\text{knee}} \cdot (2\pi(1-D)) \tag{S7}$$

$$t_{\text{swing}} = 2\pi - \phi_{\text{kneeShift}} \tag{S8}$$

$$t_{\text{half}} = \frac{t_{\text{swing}}}{\beta} \tag{S9}$$

$$\phi_{\text{flex}} = \phi_{\text{kneeShift}} + \frac{t_{\text{half}} \cdot \beta}{2} \tag{S10}$$

$$\phi_{\text{ext}} = \delta_{\text{overSwing}} - \frac{3 \cdot t_{\text{half}} \cdot \beta}{2} \tag{S11}$$

$$T_{\text{flex}} = T_{\text{ext}} = \frac{t_{\text{swing}}}{2 \cdot \beta} \tag{S12}$$

where $\phi_{\text{kneeShift}}$ is the knee activity onset, $\delta_{\phi,\text{knee}}$ is the desired phase shift between ϕ_j and knee flexion, t_{swing} is the swing duration, t_{half} is the halftime of the transient knee behavior, $\delta_{\text{overSwing}}$ is the shift between ϕ_j and knee extension onset and β is the conversion rate of the transient knee behavior (usually $2 \cdot [5..7] \cdot T_j$). The generated trajectory can be seen in Figure 4a. These equations ensure that the whole swing phase is used for knee activity and that both $\Theta_{\text{kneeAmplitude}}$ and $\Theta_{\text{kneeOffset}}$ are reached within one swing phase.

All CPG parameters p_m are modeled as a first-order differential equations to ensure smooth transitions when changing CPG parameters:

$$\dot{p_m} = \alpha_m \cdot (p_{m,\text{des}} - p_m) \tag{S13}$$

where α_m are the conversion rates for the CPG parameters and $p_{m,\text{des}}$ are the desired values for p_m. An example output for four coupled oscillators with their trajectories and a depiction of the CPG parameters described here can be seen in Figure 4a. All CPG parameters are shown in Table S2. The coupling matrix C represents a fully, bidirectionally coupled network. Full coupling ensures entrainment of the network for all initial conditions. The coupling matrix $\mathbf{C}$ and desired phase difference matrix $\mathbf{\Phi}$ are shown in Table S4.

A. Control Structure

The trajectories generated by the CPG (Equation S5 and Equation S6) are position references for PID-control loops calculating the desired joint motors currents:

$$\begin{aligned} i_{\text{hipMotor}} =& k_p \cdot (\Theta_{\text{hip,des}} - \Theta_{\text{hip}}) \\ &+ k_d \cdot \frac{\mathrm{d}}{\mathrm{d}t}(\Theta_{\text{hip,des}} - \Theta_{\text{hip}}) \\ &+ k_i \cdot \sum_j (\Theta_{\text{hip,des}} - \Theta_{\text{hip}})\Delta t \end{aligned} \tag{S14}$$

where i_{hipMotor} is the desired hip motor current, $k_{\text{p,i,d}}$ are the controller gains and Θ_{hip} is the hip angle.

$$\begin{aligned} i_{\text{feedForward}} =& \frac{c_{\text{spring}} \cdot r_{\text{cam}}^2 \cdot \Theta_{\text{hip}}}{\tau_{\text{m}} \cdot i_{\text{gear}}} \\ i_{\text{kneeMotor}} =& k_p \cdot (\Theta_{\text{knee,des}} - \Theta_{\text{knee}}) \\ &+ k_d \cdot \frac{\mathrm{d}}{\mathrm{d}t}(\Theta_{\text{knee,des}} - \Theta_{\text{knee}}) \\ &+ k_i \cdot \sum_j (\Theta_{\text{knee,des}} - \Theta_{\text{knee}})\Delta t \\ &+ i_{\text{feedForward}} \end{aligned} \tag{S15}$$

where $i_{\text{kneeMotor}}$ is the desired knee motor current, k_{p}, k_{i}, k_{d} are PID controller gains and Θ_{knee} is the knee angle. Additionally the desired knee current contains a feed-forward term, $i_{\text{feedForward}}$, that calculates the required static current to compress the knee spring based on the knee angle where c_{spring} is the knee spring stiffness, r_{cam} is the knee cam radius, τ_{m} is the knee motor torque constant and i_{gear} is the knee gear ratio. The passive knee spring provides passive impedance to the knee joint dependent on the knee angle. The feed-forward term compensates the torque the knee spring exerts on the knee joint and reduces the required controller gains to keep the PID controller stable.

S3. Feedback Mechanisms

A. Late Touchdown r_{LTD}

The CPG assumes that for $\phi = 0$ the leg is in the forward position and establishes contact with the ground. If there is no ground contact the leg should wait in this position until contact is established. Righetti et al. [32] implemented a phase deceleration during the end of swing phase. In biomechanical studies the phenomenon of swing leg retraction is reported [62] and is also reported in robots [8]. Swing leg retraction describes a period before touchdown, where the leg is already moving backwards but has not yet established ground contact. For this reason, we add a phase shift when $\phi \geq 0$ to achieve contact while allowing overswing in the leg. To achieve acceleration and deceleration in the phase oscillators we use a control term u according to [32] in Equation S2 :

$$\dot{\phi_j} = 2\pi f + \sum_{k=1}^{N} C_{jk} \cdot \sin(\phi - \phi_j - \Phi_{jk}) + u \tag{S16}$$

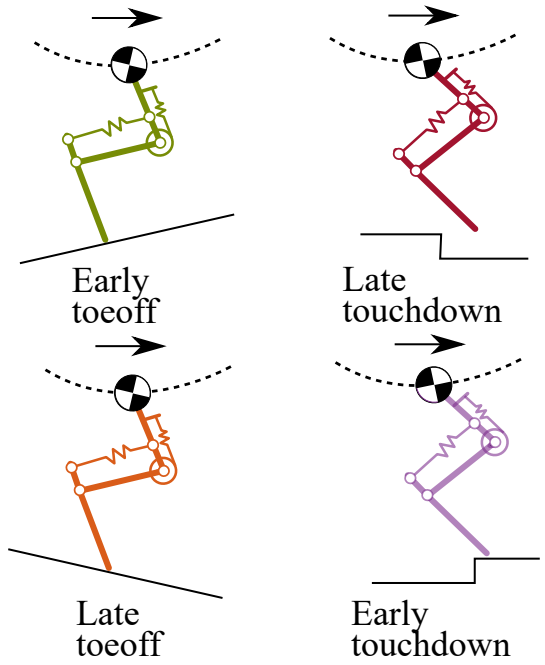


Fig. S1: **Feedback scenarios.** Depiction of four exemplary feedback scenarios where closed loop timing feedback would be active for each feedback mechanism shown in Figure 2d. Late touchdown (top left), early touchdown (top right), late toeoff (bottom left) and early toeoff (bottom right)

where

$$\begin{aligned} u =& -(2\pi f + \sum_{k=1}^{N} \alpha_{\text{dyn},j,k} \cdot C_{jk} \cdot \sin(\phi_k - \phi_j - \Phi_{jk})) \\ &\cdot (1 - (\delta_{\text{decay}})^b) \end{aligned} \tag{S17}$$

for b consecutive control loop updates of the CPG dynamics where no contact was established, where $0 \leq \delta_{\text{decay}} \leq 1$ is the base of the exponential function that decays $\dot{\phi_j}$ to zero in k steps, to decelerate the oscillator node. This deceleration can be seen in Figure 4b, in the top graphs where the slope of the oscillator phase decreases (red) until contact (gray shaded area) is established. For re-entrainment after one phase is decelerated "out of phase" we utilize the inherent coupling dynamics of the network Equation S2. Based on the weight matrix α_{dyn}, the dynamics of the network can be adjusted to enforce entrainment of the network well within one stride cycle of the robot.

B. Late Toeoff r_{LTO}

The same mechanism for late touchdown can also be applied for late toeoff. At the end of stance phase, the leg waits until the end of ground contact before initiating swing phase. This waiting period effectively also closes the loop on the timing of the knee trajectory as the knee and hip trajectories are directly coupled through the same phase reference. The late toeoff mechanism described here, however, only works for gaits with flight phase as the legs otherwise never lift off the ground because of double support during stance phase. In this case, swing phase has to be

initiated through a feed-forward timing of knee flexion. As shown in the results section, the late toeoff mechanism was not used for phase deceleration as the double stance phase of the robot prevents the legs to disengage from the ground completely and thereby prevents feedback-driven knee flexion. In Righetti et. al. [32] this mechanism was implemented successfully and is therefor also included here. For this study, we record the mismatch stemming from late toeoff but do not trigger the phase delay mechanism.

C. Early Touchdown r_{ETD}

During swing phase the legs can hit the ground prematurely, resulting in the robot stumbling. The perturbation comes from swinging the leg forwards into the ground and at the same time extending the knee. We implement a feedback loop that decreases this parasitic impact. During swing phase (Figure 2d yellow) if the leg hits the ground prematurely we want to flex the knee by an additional 5° to create more ground clearance and at the same time stop the forward swinging of the leg. We therefor disable the hip motor on impact and at the same time increase the knee flexion. The switched-off hip motor allows us to leverage the elastic leg behavior in leg length as well as leg angle direction that we presented in our previous work [8]. Impact energy that will be introduced through the premature foot impact during early touchdown will be stored in the knee and biarticular spring and will not destabilize the robot. The controller can therefor take advantage of the passive dynamic behavior of the leg as well as its elastic capabilities to improve stability. After the next phase transition to 2π the knee trajectory goes back to its normal behavior. Through the CPG coupling dynamics as well as the smoothing equations (Equation S13) the trajectory will converge back to the CPG's limit cycle without the necessity for additional feedback.

D. Early Toeoff r_{ETO}

If the leg loses contact to the ground during stance phase we want the knee to start flexing earlier to prevent further ground contact. To do so we recalculate $\Theta_{\mathrm{kneeShift}}$ (Equation S7) from the time of early toeoff to initiate early flexion and reconfigure the knee trajectory to fit the feedback timing. This mechanism provides enough ground clearance to swing the leg backward to its turning point and forwards during flight phase.

S4. Cost function

The neuroplasticity term minimizes the use of feedback by matching control dynamics to the robot's natural dynamics.

$$J_{\mathrm{feedback}} = \frac{1}{f \cdot n \cdot T} \sum_{n=0}^{4} \sum_{t=0}^{T} r_{\mathrm{ETO}} + r_{\mathrm{ETD}} + r_{\mathrm{LTO}} + r_{\mathrm{LTD}} \tag{S18}$$

where J_{feedback} is the average percentage of active feedback per step and leg, n is the number of legs, T is the evaluation time and r_{ETO}, r_{ETD} ,r_{LTO},r_{LTD} are the time vectors when the specific feedback was active for each leg.

To evaluate locomotion performance we use body position as we want the robot to cover as much distance as possible for a given CPG parameter set.

$$J_{\mathrm{distance}} = \frac{1}{x_{\mathrm{body}}} \tag{S19}$$

where x_{body} is the center of mass position in walking direction. Note that the COM position is inverted because of the minimization approach of *gp_minimize*

The contact reward term minimizes the amount of additional contact state switches (steps) per stride cycle

$$J_{\mathrm{contact}} = \frac{1}{f \cdot n \cdot T} \sum_{n=0}^{4} \sum_{t=0}^{T} \left(\left\| \frac{\mathrm{d}}{\mathrm{dt}} \mathrm{contact} \right\| > 0 \right) \tag{S20}$$

where J_{contact} is the mean amount of flight-stance changes per step, n is the number of legs, t is time, T is the evaluation duration and contact is the contact sensor data matrix for all four legs.

The periodicity term minimizes non-periodic behavior of the robot to make sure the performance of the robot does not come from undesired behavior like non-periodic jumps or skips. To do so we calculate the average distance of the maxima from the autocorrelation of the pitch angle. The pitch angle is convoluted with itself to obtain the frequency spectrum of the body pitch angle. We then compare this frequency with the actual CPG frequency, by calculating the standard deviation to determine how well the CPG and the robot's passive dynamics match. The standard deviation provides a good measure of the variation in the oscillatory behavior and is used to characterize how well the CPG dynamics fit the mechanical dynamics of the robot.

$$S_{\mathrm{pitch}} = \alpha_{\mathrm{pitch}} * \alpha_{\mathrm{pitch}} \tag{S21}$$

$$f_{\mathrm{bodyPitch}} = \| \max S_{\mathrm{pitch}} \| \tag{S22}$$

$$J_{\mathrm{periodicity}} = \sqrt{\frac{1}{N} \left(f_{\mathrm{bodyPitch}} - f_{\mathrm{cpg}} \right)^2}$$

where S_{pitch} is the frequency spectrum of the body pitch signal α_{pitch}, $f_{\mathrm{bodyPitch}}$ is the frequency of the body pitch measurement, $J_{\mathrm{periodicity}}$ is the standard deviation of the periodicity measure and f_{cpg} is the commanded CPG frequency. The body pitch term minimizes the body rotation of the robot during locomotion and ensures stable gaits and energy efficient behavior.

$$J_{\mathrm{pitch}} = \| \max(\alpha_{\mathrm{pitch}}) - \min(\alpha_{\mathrm{pitch}}) \| \tag{S23}$$

where J_{pitch} is the mean body pitch amplitude of the robot. It is calculated as the difference between mean minimum and mean maximum pitch angle of all strides during one iteration. The reward function is then calculated as the weighted sum of all the reward terms shown in Table S3. Should the robot fall before the entrainment period is over or it moves backwards, the robot is rewarded a high penalty (100) for failure. If the robot falls after the entrainment time, the performance until that point is evaluated. Here the distance reward J_{distance} is

TABLE S3: **Reward terms with estimated range and weights**

Reward factors J_j	Description	Range	Weight w_j
J_{feedback}	neuroplasticity	0 - 1	6
J_{distance}	performance	0 - $\frac{1}{10}$	9
$J_{\text{periodicity}}$	periodic behavior	0 - 2	2
J_{contact}	contact phases	0 - 5	0.5
J_{pitch}	body pitch	0 - 10	$\frac{180°}{5 \cdot \pi}$

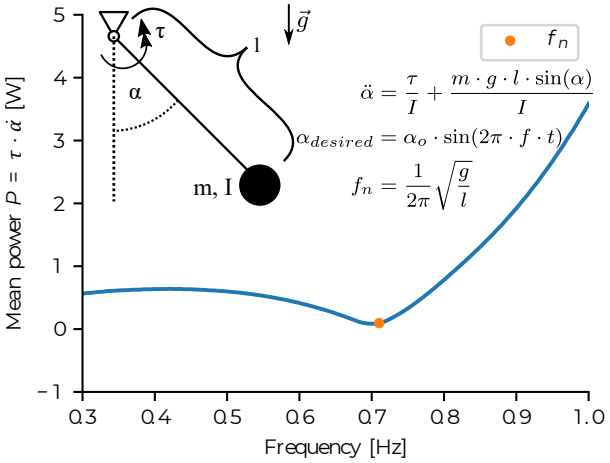


Fig. S2: **Pendulum toy example.** Toy example of the effects of matching control frequency and natural frequency in a physical pendulum. The amount of matching frequencies influences the power requirement of the actuated system. Parameters are: $m = 1\,\text{kg}$, $l = 0.5\,\text{m}$, $\alpha_0 = 30°$, $k_{\text{p,motor}} = 50$, $k_{\text{d,motor}} = 0.7$

used as a punishing reinforcement by degrading the reward signal. The reward is calculated as:

$$J = \sum_{j=1}^{N} w_j \cdot J_j \tag{S24}$$

where J is the reward signal, w_j are the reward weights and J_j are the reward factors in Equation S19 to Equation S22

S5. Toy example

To elucidate the matching of natural and controller frequency we implement a simple mathematical pendulum Figure S2. A pendulum is actuated by a torque τ to oscillate harmonically with a given frequency through a simple PD-controller. Based on the enforced oscillation frequency f from the motor compared to the natural frequency f_n of the pendulum it is easy to show the effect of mismatched frequencies. For values of f close to the natural frequency, the required power reduces. Because the natural behavior of the pendulum fits the desired motion, less control effort and motor power is required to achieve this motion.

TABLE S4: **Coupling matrix C and desired phase difference matrices Φ for different gaits**

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \tag{S26}$$

$$\mathbf{\Phi}_{\text{trot}} = \begin{bmatrix} 0 & -\pi & -\pi & 0 \\ \pi & 0 & 0 & \pi \\ \pi & 0 & 0 & \pi \\ 0 & -\pi & -\pi & 0 \end{bmatrix} \tag{S27}$$

$$\mathbf{\Phi}_{\text{bound}} = \begin{bmatrix} 0 & 0 & -\pi & -\pi \\ 0 & 0 & -\pi & -\pi \\ \pi & \pi & 0 & 0 \\ \pi & \pi & 0 & 0 \end{bmatrix} \tag{S28}$$

$$\mathbf{\Phi}_{\text{pace}} = \begin{bmatrix} 0 & -\pi & 0 & -\pi \\ \pi & 0 & \pi & \pi \\ 0 & -\pi & 0 & -\pi \\ \pi & -\pi & \pi & 0 \end{bmatrix} \tag{S29}$$

$$\mathbf{\Phi}_{\text{walk}} = \begin{bmatrix} 0 & -\frac{\pi}{2} & -\pi & -\frac{3\pi}{2} \\ \frac{\pi}{2} & 0 & -\frac{\pi}{2} & -\pi \\ \pi & \frac{\pi}{2} & 0 & -\frac{\pi}{2} \\ \frac{3\pi}{2} & \pi & \frac{\pi}{2} & 0 \end{bmatrix} \tag{S30}$$

(S31)

$$\begin{aligned} \ddot{\alpha} &= \frac{\tau}{I} + \frac{m \cdot g \cdot l \cdot \sin(\alpha)}{I} \\ \alpha_{\text{desired}} &= \alpha_0 \cdot \sin(2\pi \cdot f \cdot t) \\ \tau &= k_p \cdot (\alpha - \alpha_{\text{desired}}) + k_d \cdot \frac{\alpha - \alpha_{\text{desired}}}{\Delta t} \\ f_n &= \frac{1}{2\pi}\sqrt{\frac{g}{l}} \\ P &= \tau \cdot \dot{\alpha} \end{aligned} \tag{S25}$$

where α is the pendulum angle, τ is the motor torque, m is mass, g is gravitational acceleration, l is the pendulum length, I is the pendulum inertia, α_{desired} is the desired pendulum angle, α_0 is the oscillation amplitude, f is the oscillation frequency, t is time, k_{d} and k_{d} are the PD controller gains and P is motor power. Because this matching is non-trivial in our robotic walking machine due to highly nonlinear impedance behavior leading to nonlinear Eigenmodes and underactuation, a mathematical formulation is not possible. Alternatively, a data-driven approach can be used to approximate the performance landscape of the frequency relationship.

S6. CPG matrices

Table S4 show the coupling matrix $\mathbf{C}$ that defines the connections between CPG nodes for ϕ as in Equation S1. $\mathbf{\Phi}$ describes the desired phase differences between CPG nodes. Table S5 describes the conversion factors used in Equation S2 and Equation S13. With these factors, the convergence of the smooth transitions can be accelerated when a CPG parameter is changed. The CPG conversion factors are chosen in a way that ensures changes in trajectory-related parameters to change within one stride of the robot. Factors for frequency (α_{f}) and phase difference ($\alpha_{\text{phaseDifference}}$) are lower, so that

TABLE S5: **Conversion factors**

Conversion factor	Value
$\alpha_{\text{hipOffset}}$	4
$\alpha_{\text{hipAmplitude}}$	4
$\alpha_{\text{kneeAmplitude}}$	4
$\alpha_{\phi,\text{knee}}$	4
$\alpha_{\text{overSwing}}$	4
$\alpha_{\text{DutyFactor}}$	3
$\alpha_{\text{dyn,i,j}}$	2
α_{f}	1
$\alpha_{\text{phaseDifference}}$	1

TABLE S6: **Optimization results**

Parameter	Simulation Results	Hardware Results	Optimization Range
$\Theta_{\text{hipOffset}}$	0 °	-0.45°	[-5,5]°
$\Theta_{\text{hipAmplitude}}$	16 °	15°	[5,20]°
$\Theta_{\text{kneeAmplitude}}$	30 °	30°	[30]°
$\delta_{\phi,\text{knee}}$	0.1 [rad]	0.8 [rad]	[-0.2π,0.2π]
$\delta_{\text{overSwing}}$	0.13 [rad]	0.11 [rad]	[-0.2π,0.2π]
D_{front}	0.56	0.55	[0.3,0.6]
D_{hind}	0.56	0.55	[0.3,0.6]

transitions take several steps to change to keep the robot behavior stable.

S7. Optimization Results

Table S6 shows the comparison of optimization results from simulation as well as the parameters of the most successful hardware experiment.

S8. Gait Pattern

Figure S3 shows the gait pattern of the most successful hardware experiment. The contact data obtained from the FootTile sensors is averaged over 10 steps and displayed over a step cycle. The gait shows a slight asymmetry between legs that should contact at the same time due to minimal differences in the hip reference angles. The hind legs also show a short time where the foot loses contact right after touchdown. This can also be seen in the provided high-speed video and is due to the perturbations stemming from body pitch that are not captured in the CPG equations.

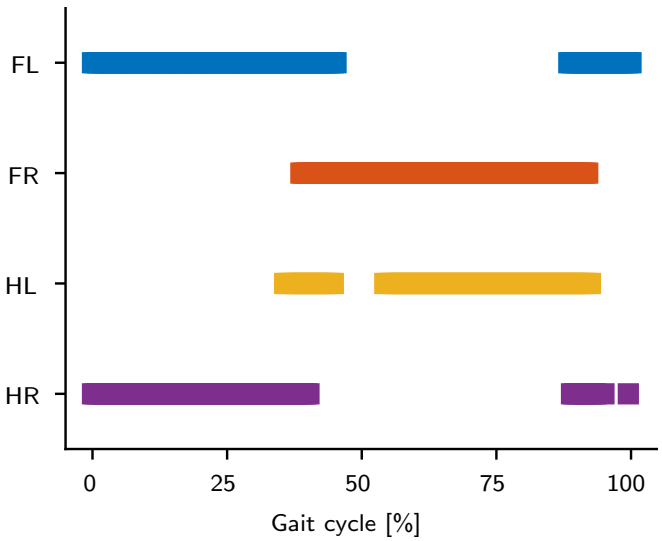


Fig. S3: **Gait pattern.** Emerging gait pattern after optimization in hardware. The gait that emerged is a trot gait. Data shown here is averaged over 10 steps. In both hind legs a small fraction of time is visible where the legs lose contact to the ground due to the pitching body.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- video1.mp4
- video2.mp4
- nreditorialpolicychecklist.pdf