Dynamic Resonance Frequency Identification for Economic Insect-scale Legged Robot Locomotion

Luca Russo*, Elyssa Chandler†, Kaushik Jayaram‡, Amit Ranjan Trivedi*
*AEON Lab, University of Illinois at Chicago, Chicago, IL, USA {lrusso5, amitrt}@uic.edu
†Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, elyssac@andrew.cmu.edu
‡University of Colorado at Boulder, Boulder, CO, USA, Kaushik.Jayaram@colorado.edu

2024 6th International Conference on Control and Robotics (ICCR) | 979-8-3315-1815-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCR64365.2024.10927506

***Abstract*—Advances in robotics and manufacturing processes have enabled the development of extremely small robotic devices approaching the size of insects. In this domain, legged microrobots (*microbots*) offer numerous potential applications and characteristics to explore. However, controlling such small multi-legged robots presents significant challenges in achieving the desired behavior. Primarily, due to the robot's small size, it can only operate with a tiny battery, therefore, an extremely computationally efficient controller is needed. Tiny robots are also susceptible to damage and control methodologies are needed that also ensure longevity. This paper presents a novel approach for creating a control algorithm for a multi-legged system that dynamically identifies and operates a microbot at its resonance frequency of movement. At the resonance frequency, a microbot's leg oscillations achieve maximum amplitude with minimal energy, resulting in optimal locomotion efficiency. By imposing a sinusoidal control for actuating each of the robot's legs, a two-part controller is developed. The controller is trained using soft actor-critic-based reinforcement learning on a custom model of the mClari microbot. A successful simulation-based demonstration of the learning-based controller is shown by varying the underlying surface (such as wet, soft, etc.) of the microbot where the controller dynamically identifies and switches to the corresponding resonance frequency, achieving efficient and adaptive locomotion.**

***Index Terms*—Microbot, Locomotion, Resonance Frequency, Energy Efficiency, Reinforcement Learning, Actor-Critic.**

I. Introduction

Legged micro-robots (i.e., microbots), tiny devices of millimeter size, have recently been presented. For example, microbots such as the HAMR-VI [1] and the cockroach-inspired mClari [2] have been developed. These microbots, comparable in size to a penny, offer intriguing application prospects. For instance, in agriculture, cameras embedded on microbots can locate infected plants to prevent disease spread. In industrial plants, microbots can navigate through constricted spaces such as gas pipelines to identify gas leaks. During earthquakes, autonomous microbots can assist in search and rescue operations by navigating through building debris. Within homes, microbots can actively monitor for intruders.

Supporting these microbots, there has been significant progress in maximizing the area/energy efficiency of on-board robotic workload processing. [3], [4] demonstrated a novel compute-in-memory approach to remarkably minimize the necessary energy for robotics algorithms such as visual odometry. Shukla *et al.,* [5]–[7] demonstrated novel computing architectures and technologies to enable robotic reasoning for localization and path-planning under uncertainties. [8]–[10] demonstrated ultra-low-power solutions to check the reliability of sensing and to take advantage of generative AI methods to minimize the need for complex sensors such as LiDAR. Suleiman *et al.,* [11], [12] demonstrated chip designs operating within 2 mW power to support a variety of robotic navigation algorithms.

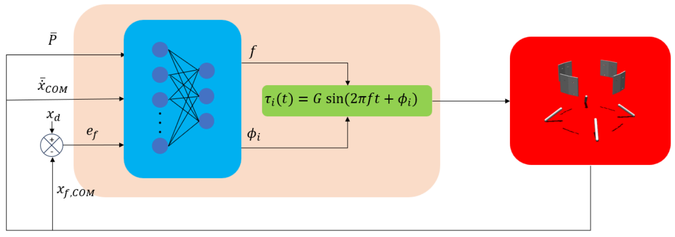

Fig. 1: Schematic of the controller logic and system model: Proposed control framework identifies the frequency and phase difference angles to compute actuation signals, which are fed into the system model for dynamic identification and operation at resonance frequency.

Despite this significant potential, effectively controlling microbot's locomotion presents several challenges. Since a microbot can only carry the payload of a tiny battery, minimizing power dissipation for its locomotion becomes quite critical. A microbot must also operate with only very few lightweight sensors and actuators since each not only demands more power but also increases the size and weight of the vehicle. Moreover, the energy efficiency of their locomotion depends on several time-varying factors which also vary from device-to-device. For instance, these robots typically rely on piezoelectric actuators, which exhibit inherent frequency-dependent characteristics. Similarly, the movement of these tiny devices is driven by an actuation kinematic chain composed of compliant links and flexure joints, where manufacturing imperfections can significantly affect the frequency response from one device to another. Devices like mClari and HAMR-VI have even more intricate coupled systems, with each leg controlling two different degrees of freedom (DoF)–lift and swing–making the system highly interdependent.

Addressing the above challenges, this work develops a novel *physics-informed* control mechanism that dynamically identi-

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on May 17,2025 at 03:08:36 UTC from IEEE Xplore. Restrictions apply.

fies and operates a microbot at its resonance frequency. At this frequency, the microbot moves with maximum amplitude and minimal energy input, as the system's natural frequency aligns with the frequency of external forces, resulting in optimal efficiency and performance. The resonance frequency is also influenced by the underlying surface due to variations in compliance and friction. On softer surfaces, the microbot's legs may sink slightly, lowering the resonance frequency due to altered leg-surface interaction stiffness. Conversely, harder surfaces increase rigidity, raising the resonance frequency. Additionally, surface texture and friction affect energy transfer and damping, further modifying the optimal resonance frequency for efficient locomotion.

Considering such temporal variations in the resonance frequency during locomotion, the proposed controller dynamically identifies and adjusts the locomotion frequency to maintain optimal performance across different surfaces. We specifically studied the controller on mClari, a four-legged microrobot developed by the University of Colorado Boulder [2]. The remaining paper is organized as follows: Sec. II presents the controller logic, Sec. III details the simulation model used for training and testing, Sec. IV discusses the reinforcement learning architecture, Sec. V presents the results, Sec. VI provides conclusions, and Sec. VII outlines future works.

II. Controller Objectives and Design

mClari [2] is a four-legged microrobot actuated by eight piezoelectric actuators, each driven by phase-offset sinusoidals:

$$V_i(t) = V_{b,i} \sin(2\pi f_i t + \phi_i). \tag{1}$$

Each actuator is controlled by three variables (V_i, f_i, ϕ_i), thus totaling 24 controllable variables. Considering the need for a resource-efficient on-board controller, to simplify this expensive control space, we fit the maximum amplitude of applied voltages ($V_{b,i}$) to a constant across all actuators. To emulate typical locomotion control in robots and animals, we drive all the legs at the same frequency. Prior research on similar robots like the HAMR-VI [13] have shown effective movement through only phase adjustments, which our preliminary tests (see Sec. V) also confirm. Therefore, all directional movements were achieved by adjusting phase shifts while operating the robot at resonance frequency for energy efficiency. On this reduced control space, mClari's controller was designed to achieve three key objectives: (i) following the desired path, (ii) ensuring longevity by avoiding excessive leg stretches that could cause long-term damage, and (iii) maintaining energy efficiency given its small battery.

Inspired by previous works on legged robots [14], we employed reinforcement learning (RL)-based control, incorporating both endogenous signals from the robot (power consumed by actuators, robot velocity, and center of mass position) and an exogenous component (desired center of mass position). Operating on these signals, a deep neural network (DNN)-based reinforcement learning agent extracts insights from the robot's operating state and dynamically tunes a reduced state of actuation variables.

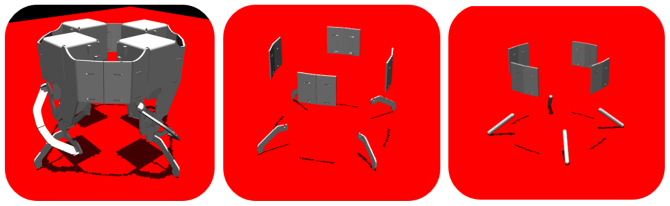
Fig. 2: Phases of the simulation models. From left to right: the initial complete with all the robot's links and joints, the first simplification with just the leg outputs and the connections between legs and the second simplification where the leg outputs have been substituted with capsules.

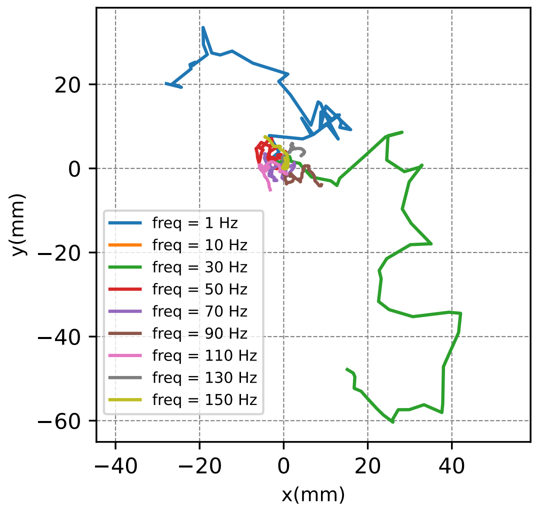

Fig. 3: Trajectories by the robot after 60 seconds of simulation for different frequencies with intra-leg phase difference of $pi/2$ and no inter-leg difference (triple $(0, 0, 0)$).

Based on previous work on microrobots [15], which showed that two-layer NNs are sufficient for effective learning and adaptation in similar environments, we used a similar NN structure for our controller. Figure 1 provides a schematic of the developed controller logic. Using such learning-based control structure, the controller also selects the desired group of phase angles to follow the intended direction, functioning like a finite state machine (FSM) by choosing among fixed angle combinations while aiming to reach the target within a specific timeframe along with maximizing energy efficiency.

III. Simulation Model and Characterizations

A. Simulation Model of mClari Microbot

To evaluate the controller, we created a custom environment using the mClari robot, developed by the University of Colorado Boulder [16], as a starting point. Utilizing the open-source physics engine MuJoCo [17], we modeled and visualized the mechanical system. Due to the small dimensions of the robot, we adopted MMGS (millimeter-gram-second) units instead of the standard MKS (meter-kilogram-second) to avoid

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on May 17,2025 at 03:08:36 UTC from IEEE Xplore. Restrictions apply.

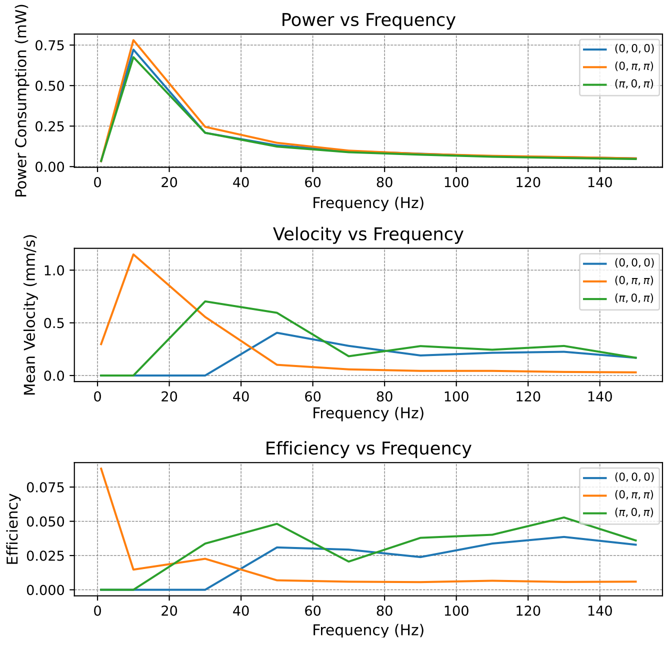

Fig. 4: Power consumption, mean velocity, and efficiency of robot movement for different frequencies with fixed inter-leg phase shifts triples of (0,0,0), (π,π,0) and (π,0,π).

numerical issues with very small component dimensions. All forces and environment dimensions were scaled accordingly.

Our modeling process followed a three-step approach:
1. Initial Model: Using CAD files, we obtained mesh files for modeling links and joints. However, closed-chain kinematics in each leg caused issues, as MuJoCo works with open kinematic chains, requiring modifications that led to a malfunctioning model.
2. Simplified Features: We focused on the leg end effector and top connection to better simulate the quadruped's dynamics, addressing the kinematic challenges.
3. Optimized Model: Inspired by the ant model in MuJoCo, we replaced leg mesh files with capsules, reducing computations and improving contact point identification and precision.

Figure 2 shows the three models used, from left to right, from the most complex to the simplest.

B. Actuator Design

For the actuator design, we started with the physical results from Doshi et al. [18], deriving the Bode plot of the frequency response for each leg's kinematic chain. Their findings indicate that each actuator chain behaves like a second-order system, allowing us to identify the inertia, damping, and stiffness coefficients for the actuation motor. According to the prior work, for each actuator, the frequency behavior of the amplitude of the leg's oscillation in its respective direction, for a fixed voltage, follows the transfer function:

$$H(s) = \frac{3200}{s^2 + 8s + 1600} \tag{2}$$

To simplify the implementation, we replaced the piezoelectric actuators with motors to control each leg's degrees of freedom (DoF). Based on the transfer function, we characterized the actuator's armature, stiffness, and damping coefficients.

C. Surface Definitions

Finally, we designed various surfaces to evaluate the control algorithm for dynamically finding the optimal frequency. Six types of surfaces were implemented based on two classifications: surface texture and slipperiness level. The surface texture can be either rough or plain (with or without height field), while the slipperiness level can be categorized as normal, wet, or extra dry. The surface texture was obtained using random Gaussian noise $N(0, 0.1)$, while the slipperiness level was determined through three different friction coefficients: sliding (SC), torsional (TC), and rolling (RC). The table below lists the values used for the three surfaces:

TABLE I: Friction coefficients for the three slipperiness

Surface	Sliding (SC)	Torsional (TC)	Rolling (RC)
Normal	1.0	0.005	0.0001
Wet	0.1	0.1	0.1
Extra Dry	1.0	1.0	1.0

IV. Reinforcement Learning

For RL-based resonance frequency identification across different terrains, we created a custom gym environment [19]. This environment defined all the main RL components, such as the action space, observation space, and rewards. We then used the Stable Baselines3 (sb3) library [20] to train the agent using the Soft Actor-Critic (SAC) algorithm. Pseudo-code of RL-based resonance identification is given below and subsequently, the elements of the gym environment are defined:

Algorithm 1 Reinforcement learning step

```
1: compute initial observation
2: while terminated == TRUE do
3:     compute action
4:     execute step
5:     compute reward, new observation
6:     episode reward += $\gamma$*reward
7: end while
8: return episode reward, terminated
```

Observation Space: Due to the robot's small size, only a few sensors can be placed on it. Therefore, we decided to use just four state space variables. The first is the overall power consumed by the actuators, obtained from MuJoCo or, in a real scenario, from the piezoelectric actuators. It is calculated by multiplying the torque of each actuator by the angular velocity of the respective joint and by taking the mean of the norm of all the power values. The second value is the mean velocity of the center of mass of the robot, determined by the difference in the position of the robot's center before and after a step. The last two values are the errors along the x and y axes between the target position of the robot and its actual position.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on May 17,2025 at 03:08:36 UTC from IEEE Xplore. Restrictions apply.

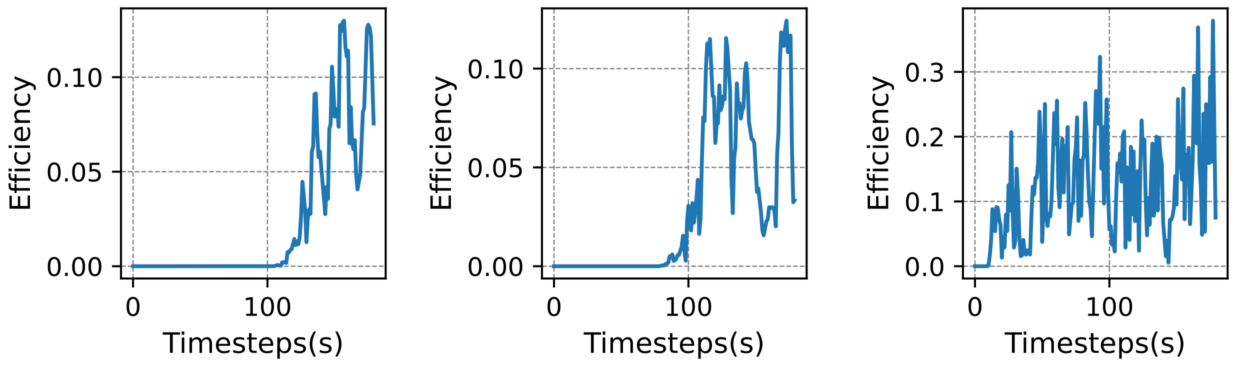

Fig. 5: Results of the gait efficiency of the simulated robot after 3 minutes of simulation on the tree plain terrains.

Action Space: We worked with two normalized action variables: one for the frequency and another for selecting the phase angles group. Frequency changes were constrained to be gradual to avoid large jumps that could damage the robot. Similarly, the second action variable selects the optimal angle combination. Once both frequency and angles are computed, the net actuation signals can be obtained.

Agent: The agent is a two-layer neural network with 10 nodes in the first layer and 3 nodes in the second layer. It takes the 4 observation space values as input and computes two normalized float numbers as output. The agent is trained using the SAC algorithm implemented by the sb3 library [20].

Action Step: Once the agent determines the action to be performed (i.e., the frequency and phase alteration), the initial position of the robot is recorded, and the robot is simulated for one second using MuJoCo. At each timestep, the power consumed by the actuators is registered. At the end of that period, the final position of the robot is obtained along with the mean power consumed by all the actuators. The observation states are then updated, and the reward is computed.

Reward Function: In developing the reward function, two main objectives were kept in mind: maximizing the robot's movement speed while minimizing overall energy consumption and reaching the target position. For the first objective, we focused on maximizing the overall system efficiency η (see 3). For the second objective, we aimed to minimize the position errors, calculated as the distance between the desired and the actual position of the robot's center of mass.

Termination: The termination signal indicates when an episode has ended, used for both training and simulation. We used three variables for termination: two boolean variables, *terminated* and *truncated*, and a dictionary, *info*. *Terminated* states whether the episode has ended, *truncated* indicates whether the termination is due to reaching the maximum number of steps or other causes, and *info* contains information related to the end of the episode. The maximum length of an episode was set to 180 steps (3 minutes). Termination conditions include the robot falling (resulting in a penalty) and reaching the chosen target.

V. Simulation Results

Once the system model was complete and the actuators fully characterized, we aimed to determine if the results from [13] could be repurposed and validated for our mClari robot simulation. To fully understand the behavior influenced by phase angles and frequencies, we fixed the amplitude of all actuation signals and focused solely on the eight phase angles ϕ_i. Initially, we reduced the phase differences from eight to four by enforcing a fixed and constant phase shift between the swing and lift movements in each leg (i.e. *intra-leg* phase shift). Our tests indicated that a phase shift of $\pi/2$ provided the best results, resulting in a circular motion at the tip of the robot leg in the swing and lift degree-of-freedom (DoF) plane, leading to softer contact. Next, we fixed the phase angle of the first leg to zero and defined other angles relative to it (i.e. *inter-leg* phase shifts). We identified three additional angles by defining three characteristic angles: $(0, \pi/2, \pi)$, and found all possible permutations, producing 27 triples in total.

For each triplet, we ran eight simulations of the MuJoCo model described in Sec. III for 1 minute, varying the frequency from 1 Hz to 150 Hz ([1, 20, 40, 60, 80, 100, 120, 150]). We recorded the position of the robot's center-of-mass and plotted the trajectory over 1 minute. The results, similar to those presented in Fig. 3, showed a wide range of trajectories by fixing the phase angle differences and varying the frequency. This indicated that frequency and phase angles are interconnected, and by choosing the correct tuple (frequency, phase angles), any type of movement and direction can be achieved. Likewise, to maximize the energy efficiency of movements, we used a similar approach. Using the same 27 angle groups, we ran eight 10-second simulations, one for each frequency. For each frequency, we averaged the power consumed by each motor and the mean velocity of the robot. We then evaluated an efficiency parameter η defined as:

$$\eta = \frac{mgv}{\sum_{i=1}^{8} \bar{P}_i} \tag{3}$$

Figure 4 presents the plots of power consumed, mean velocity, and efficiency with respect to frequency for three groups of angles. The results indicated that while power consumption was nearly the same across different angle combinations,

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on May 17,2025 at 03:08:36 UTC from IEEE Xplore. Restrictions apply.

velocity and efficiency had specific frequency values that maximized performance (resonance frequency). These optimal frequencies depended on the chosen angle group. Furthermore, these results were obtained on a plain surface, but different surfaces exhibited different resonance frequencies.

The previous setup was used to train the neural network controller using the Soft Actor-Critic (SAC) algorithm implemented in the sb3 library [20]. After tuning the hyperparameters (learning rates, size of the NN layers, etc.), we found that training the algorithm for 500,000 steps enabled the system to learn the resonance frequency for each terrain. Each time an episode ended, the new one featured a different terrain for the robot to navigate and a different target to reach. This approach aimed to train the neural network to adapt quickly to changes, improve system robustness, and promote generalized behavior by avoiding focus on a single terrain type. After the training phase, we tested the resulting model on all six different terrains for an entire episode. We recorded the efficiency of the robot's movement and plotted it over time. Figure 5 shows the obtained results for three of the tested terrains, the plain ones. It can be seen that the agent can identify the movement that leads to high efficiency, and the research is quite fast, considering that it usually takes around 1 to 2 minutes of simulation.

VI. Conclusions

This paper presents a novel approach to control microbots using just two variables: frequency and phase shifts. These two parameters enable full control over the robot's position and direction, and by identifying the resonance frequency, the robot's efficiency can also be optimized. A RL-based control setup was presented that under changing terrains dynamically identifies the resonance frequency to maximize the energy efficiency of movements.

VII. Future Works

Several future research directions remain to be explored. Firstly, the current work is confined to simulations. Future research could explore advanced algorithms or hybrid approaches to improve trajectory tracking performance in real systems. Secondly, extending the results to a swarm of microbots is a promising avenue. By enabling robots to operate independently and subsequently merge their findings, the computational cost of determining optimal frequencies could be significantly reduced. This approach could enhance the efficiency and scalability of microbot systems, allowing for more complex and adaptive behaviors in swarm applications.

Acknowledgement: This work was supported by National Science Foundation (NSF) Grant # 2235207.

References

[1] K. Jayaram, J. Shum, S. Castellanos, E. F. Helbling, and R. J. Wood, "Scaling down an insect-size microrobot, hamr-vi into hamr-jr," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 305–10 311.

[2] H. Kabutz, A. Hedrick, W. P. McDonnell, and K. Jayaram, "mclari: a shape-morphing insect-scale robot capable of omnidirectional terrain-adaptive locomotion in laterally confined spaces," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 8371–8376.

[3] P. Shukla, A. Muralidhar, N. Iliev, T. Tulabandhula, S. B. Fuller, and A. R. Trivedi, "Ultralow-power localization of insect-scale drones: Interplay of probabilistic filtering and compute-in-memory," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 30, no. 1, pp. 68–80, 2021.

[4] P. Shukla, S. Sureshkumar, A. C. Stutts, S. Ravi, T. Tulabandhula, and A. R. Trivedi, "Robust monocular localization of drones by adapting domain maps to depth prediction inaccuracies," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[5] P. Shukla, S. Nasrin, N. Darabi, W. Gomes, and A. R. Trivedi, "Mc-cim: Compute-in-memory with monte-carlo dropouts for bayesian edge intelligence," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 2, pp. 884–896, 2022.

[6] N. Darabi, P. Shukla, D. Jayasuriya, D. Kumar, A. C. Stutts, and A. R. Trivedi, "Navigating the unknown: Uncertainty-aware compute-in-memory autonomy of edge robotics," in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2024, pp. 1–6.

[7] C. Lee, L. Rahimifard, J. Choi, J.-i. Park, C. Lee, D. Kumar, P. Shukla, S. M. Lee, A. R. Trivedi, H. Yoo *et al.*, "Highly parallel and ultra-low-power probabilistic reasoning with programmable gaussian-like memory transistors," *Nature Communications*, vol. 15, no. 1, p. 2439, 2024.

[8] S. Tayebati, T. Tulabandhula, and A. R. Trivedi, "Sense less, generate more: Pre-training lidar perception with masked autoencoders for ultra-efficient 3d sensing," *arXiv preprint arXiv:2406.07833*, 2024.

[9] N. Darabi, S. Tayebati, S. Ravi, T. Tulabandhula, A. R. Trivedi *et al.*, "Starnet: Sensor trustworthiness and anomaly recognition via approximated likelihood regret for robust edge autonomy," *arXiv preprint arXiv:2309.11006*, 2023.

[10] A. Shylendra, P. Shukla, S. Mukhopadhyay, S. Bhunia, and A. R. Trivedi, "Low power unsupervised anomaly detection by nonparametric modeling of sensor statistics," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 8, pp. 1833–1843, 2020.

[11] A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, and V. Sze, "Navion: A 2-mw fully integrated real-time visual-inertial odometry accelerator for autonomous navigation of nano drones," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 1106–1119, 2019.

[12] ——, "Navion: A fully integrated energy-efficient visual-inertial odometry accelerator for autonomous navigation of nano drones," in *2018 IEEE symposium on VLSI circuits*. IEEE, 2018, pp. 133–134.

[13] B. Goldberg, N. Doshi, and R. J. Wood, "High speed trajectory control using an experimental maneuverability model for an insect-scale legged robot," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3538–3545.

[14] G. A. Castillo, B. Weng, W. Zhang, and A. Hereid, "Reinforcement learning-based cascade motion policy design for robust 3d bipedal locomotion," *IEEE Access*, vol. 10, pp. 20 135–20 148, 2022.

[15] B. P. Duisterhof, S. Krishnan, J. J. Cruz, C. R. Banbury, W. Fu, A. Faust, G. C. de Croon, and V. J. Reddi, "Learning to seek: Autonomous source seeking with deep reinforcement learning onboard a nano drone microcontroller," *arXiv preprint arXiv:1909.11236*, 2019.

[16] H. Kabutz and K. Jayaram, "Design of clari: A miniature modular origami passive shape-morphing robot," *Advanced Intelligent Systems*, vol. 5, no. 12, p. 2300181, 2023.

[17] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012.

[18] N. Doshi, K. Jayaram, S. Castellanos, S. Kuindersma, and R. J. Wood, "Effective locomotion at multiple stride frequencies using proprioceptive feedback on a legged microrobot," *Bioinspiration & biomimetics*, vol. 14, no. 5, p. 056001, 2019.

[19] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016. [Online]. Available: https://arxiv.org/abs/1606.01540

[20] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann, "Stable baselines3," https://github.com/DLR-RM/stable-baselines3, 2021.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on May 17,2025 at 03:08:36 UTC from IEEE Xplore. Restrictions apply.