Pupper: An Open-source Quadruped Robot and Curriculum for AI Robotics Education*

1st Given Name Surname
dept. name of organization (of Aff.)
Stanford University
email

2nd Given Name Surname
dept. name of organization (of Aff.)
Stanford University
email address or ORCID

***Abstract*—Lack of reliable, low-cost, and accessible hardware poses significant challenges for robotics and AI education, while hands-on experience has been demonstrated to enrich the learning experience. In order to address this challenge, we developed a low-cost and open-source quadruped robot, Pupper, and comprehensive AI curriculum. The robot features torque-controllable brushless motors with high specific power that enable testing of impedance and torque-based machine learning and optimization control approaches. Pupper can be built from the ground up in under 8 hours for a total cost under 2000, with all components either easily purchased or 3D printed. The accompanying curriculum goes through hardware building, basics in robotics (PID, FK, IK), simulation, and robot learning (RL). By the end of the curriculum, the students are able to train an end-to-end locomotion policy and deploy it on the physical Pupper robot.**

***Index Terms*—Education, Robot Learning, Legged Locomotion**

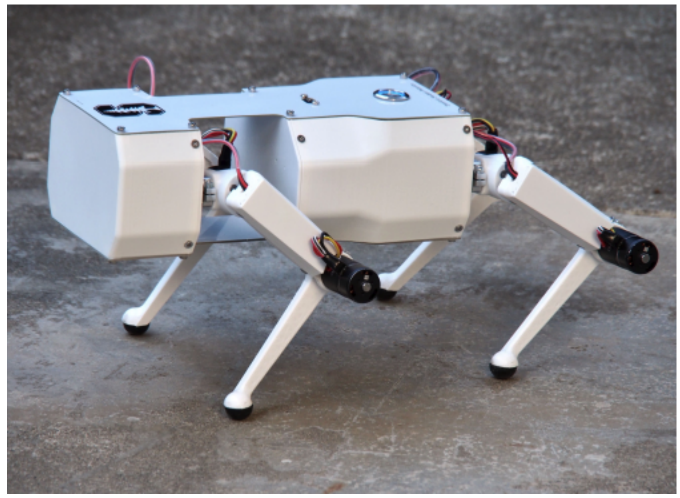

Fig. 1. **Pupper Robot:** a low-cost, agile quadruped designed for teaching an AI/Robotic curriculum and for studying novel reinforcement learning techniques

I. Introduction

The Pupper robot is an open-source, full-stack robotics platform that is designed to teach fundamental robotics concepts in a cost-effective and accessible manner while still meeting state-of-the-art research demands. With the fast-paced iteration of the AI/robotics field, it is essential to teach novel methods in fields such as task-planning, sim-to-real, and reinforcement learning in an educational setting while interfacing with actual hardware. Taking inspiration from other high-performance quadruped robots such as MIT Cheetah, Unitree GO1, and Anymal, we sought to create a more affordable quadruped robot that can support state-of-the-art research while meeting educational demands. The Pupper robot and the accompanying curriculum was designed in tandem between students from the Stanford Robotics club and the non-profit organization Hands-On Robotics.

II. Background and Motivation

A. Current state of AI Robotics Courses

The absence of accessible, open-source hardware poses challenges for successful AI and robotics education. Characterizing how hardware components affect a robot's overall performance is critical to understanding its fundamental strengths and limitations. Also, rapid progress in AI research necessitates teaching the latest techniques to equip students with relevant skills. Unfortunately, many open-source robot platforms do not easily integrate with the most recent developments in AI, posing a barrier to effectively teaching about the field's dynamic landscape.

To our knowledge, few classes teach both AI and hands-on robotics curriculum in the same course. We examined the curriculum for the top 10 ranked engineering schools according to US News for 2024: Massachusetts Institute of Technology, Stanford University, University of California, Berkeley, Purdue University, Carnegie Mellon University, Georgia Institute of Technology, California Institute of Technology, University of Michigan, University of Texas at Austin, and Texas A&M. We found that these universities commonly teach hands-on robotics classes (which we define as courses in which students build their own hardware). However, within these classes it is very rare for students to learn and deploy machine learning models beyond simple API calls to MobileNet and other older computer vision libraries. Typically, computer vision models are used for autonomous navigation for vehicles such as TerraBots and MIT Racecar [citation]. We found that rarely, students are given the opportunity to use mobile manipulators, particularly for end-of-class projects. However, manipulators are expensive, never constructed by the students, and not often integrated into the core curriculum. We were unable to find a

Identify applicable funding agency here. If none, delete this.

single class teaching applications of reinforcement learning or LLMs for deployment on hardware students build. We believe it is essential for these topics to built into the core curriculum of AI and robotics classes through labs - not just final projects.

Jaden Compare to CS274A, talk about why stanford is a good point of comparison

B. Current State of the Art Robots

- **Gabrael**
- Comparison to other robots, MIT Cheetah, Solo, Anymal, etc. Focus on speed, weight, cost, time to build, etc. (not open source)

Reference	Platform	Robot Type	Cost (USD)	Weight (lbs)	Fully Open Source	Open Curriculum
1	Ours	Quadruped	2376	7	✓	✓
2	SOLO 12DoF	Quadruped	2800	✓		
2	Mini Cheetah	Quadruped	2800	✓		
2	Doggo	Quadruped	2800	✓		
2	Unitree Go1	Quadruped	2800	✓		

TABLE I
CAPTION

C. Current State of AI Robotics Educational Platforms

- **Liana**
- Comparison to other robotics and AI education methods, closed-source and expensive materials, hardware-less courses, etc.

Reference	Platform	Robot Type	Cost (USD)	Weight (lbs)	Fully Open Source	Open Curriculum
1	Ours	Quadruped	2376	7	✓	✓
2	MIT Racecar	4 wheeler	2800	✓		
2	Turtlebot	4 wheeler	2800	✓		

TABLE II
CAPTION

III. OUR PLATFORM

A. Robot Design

- **Gabrael**
- Labeled CAD fig
- Materials and fabrication
- system architecture

B. Robot Characterization

1) Motor Current and Torque Relationships: For Pupper to be used as a platform for researching and deploying novel learning and optimization methods, accurate physical robot models are necessary to calculate motor current commands. To understand the actuator limits, torque-current relationship, and control bandwidth, the Pupper actuator was tested in a dynamometer. The results are summarized in Fig. 2. When doing positive work, the actuator's peak torque was 1.8Nm, and while doing negative work 3.2Nm. The maximum continuous measured torque was 1.0Nm.

Motor friction was modeled well by Coulomb and damping terms.

$$\tau_f = -0.021\text{Nm}\,\text{sgn}\,(\omega) - 0.045\frac{\text{Nms}}{\text{rad}}\omega - 10.0\,\text{sgn}(\omega)|\tau_m| \quad (1)$$

where τ_f is the total friction in Nm, ω is the output velocity in $\frac{\text{rrad}}{\text{s}}$, and τ_m is the motor torque at the gearbox input in Nm. In this model, an R^2 value of over 0.999 was achieved. The motor torque is given by

$$\tau_m = 0.0069\frac{\text{Nm}}{\text{A}}i \quad (2)$$

where i is the motor current in A. The output current is modeled as

$$\tau_{output} = 36\tau_m + \tau_f \quad (3)$$

where τ_{output} is the output torque, and the factor of 36 comes from the 36:1 reduction ratio.

To predict the motor current necessary to achieve desired torque commands, the friction model was inverted. Arbitrary torques can be commanded within 1% error at constant motor velocities. However, stiction cannot be predicted because the inverted model only corrects for velocity-dependent friction, leading to 28% torque error when the actuator has zero velocity.

Rotor inertia was estimated using the torque-current relationship by meeasuring the output acceleration at fixed torques. Figure 3 shows a bode plot determining the bandwidth of the actuator, where sinusoidal currents were commanded and the magnitude of the output torque was measured as a function of frequency.

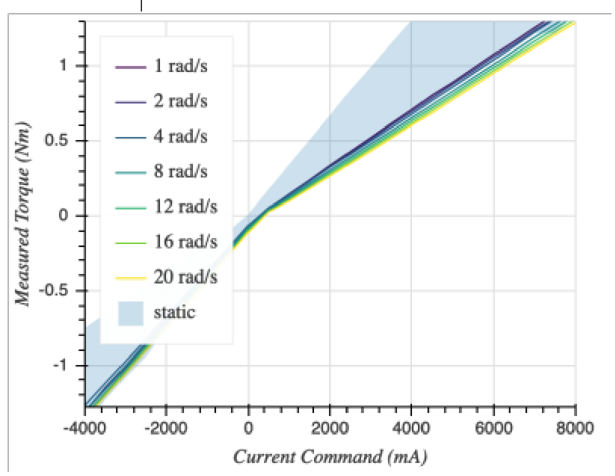

Fig. 2. Measured torque versus commanded current across several motor velocities. The asymmetry between the positive and negative current cases is due to nonlinear friction in the actuator.

2) Maximum Kinetic Energy of Joints: For Pupper to be used in educational settings, it must be safe for students to use in a hands-on manner, meaning that the maximum kinetic energy of the robot should be limited.

Gabrael to fill in maximum KE calculation. Cite something saying that is less than the other robots as a comparison

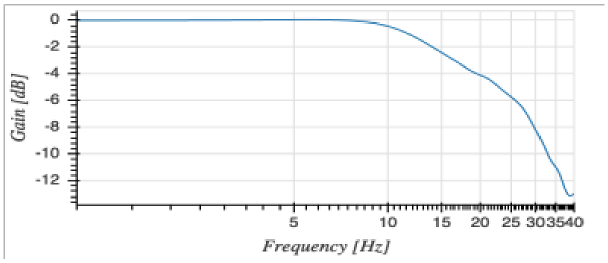

Fig. 3. Actuator frequency response. A sinusoidal current command ranging from 0A to 5A was commanded while the frequency increased from 0Hz to 40Hz. The gain of the resulting torque was measured.

3) Repeatability: For Pupper to be used in educational settings where novel learning and optimization methods are tested and compared against each other, Pupper must be reproducible across various builds. To measure this, a baseline task was used, where each robot would complete the sprint independently, and the results compared to give a marker of repeatability.

Task Overview: Three total robots were built, at different times at different universities. One was built by the authors, and the two others were built by undergraduate engineering students at Massachusetts Institute of Technology and Worcester Polytechnic Institute in under a day. To study repeatability, a simple Sprint task was designed. As outlined in Fig. 4, Pupper will traverse an unobstructed 5m course as fast as possible. This task will test Pupper's speed of locomotion under standardized settings, a long-standing goal of the legged-robotics community. Achieving high-speed locomotion requires tightly coordinating full-body motion while managing ground impacts and destabilizing perturbations. To standardize the robot's gait during this task, a trotting controller that generates foot position targets as a function of time and desired velocity in the horizontal plane was implemented. The architecture is similar to the position-based controller in Stanford Doggo (cite) and the Foot Trajectory Generator (FTG) architecture formalized in (cite).

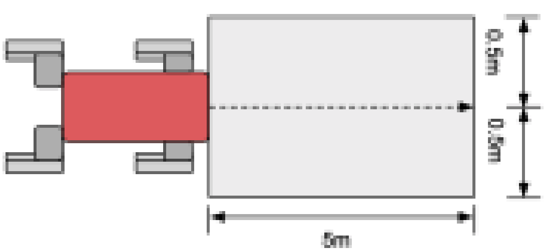

Fig. 4. The Sprint task requires the robot to travel five meters forward as fast as possible.

Task Results: The three robots, each built at a different university, completed the sprint task with an average speed of 0.66 m/s, and a standard deviation of 0.025. Fig. 5 illustrates the interquartile regions for these sprint speeds across the trials for each robot. Fig 6 compares the total electrical power of each of the robots when performing the sprint task. In terms of the sprint speeds and the sprint electrical power, all robots exhibited repeatable results within small error of each other.

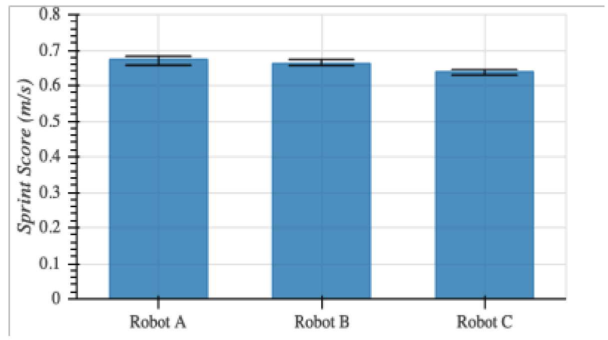

Fig. 5. Comparison of sprint speeds between three different Pupper robots built at three different institutions. The stem and whiskers indicate the interquartile range of the sprint results. All three robots recorded repeatable sprinting speeds within low relative error of each other.

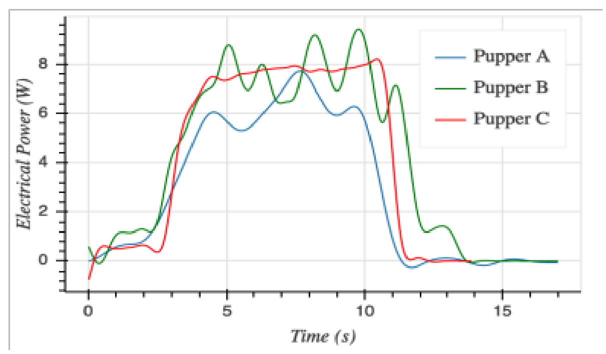

Fig. 6. Comparison of total motor electrical power across the three Pupper robots for the Sprint task.

C. Applications in Reinforcement Learning, Computer Vision, and Large Language Models

Jaden - show different cadence with different speeds

Pupper is a capable platform for deployment of SoTA robot learning methods, in particular for students with little past experience. Pupper has a custom hardware setup for Luxonis OAK-D lite depth cameras, and students have applied learning-based techniques to enable Pupper to follow balls, and detect gestures - all using onboard processing. Pupper also has a simple API to query Large Language Models (LLMs), allowing control of Pupper using natural language input or voice commands. By the end of our curriculum, students are able to train end-to-end locomotion policies, implement object tracking, and solve task planning problems with LLMs on the physical Pupper robot.

Pupper is a paricularly capable platform for teaching reinforcement learning. We offer a physically accurate Unified

Robot Description Format (URDF) model for rapid experimentation and straight forward integration with robot learning environments such as Nvidia Isaac Gym [reference]. Using Isaac Gym, agile locomotion policies can be trained in under 20 minutes.

We find that Pupper is sufficiently performant to demonstrate the effects of different training methods. In [ref figure] we show that domain randomization is essential for walking performance on Pupper. By guiding students through lab with and without randomization, students are able learn the importance of this technique for deployment of learning algorithms on physical robots. Pupper also demonstrates emergent locomotion behavior. In [ref figure] we show that Pupper is capable of learning different locomotion styles when trained with different desired velocity commands.

Jaden TODO - add bar chart for cadence vs velocity, add figure for speed tracking with and without doman randomization

D. Curriculum Design

We determined four fundamental AI Robotics topics: control, kinematics, locomotion, and hardware familiarity. Through a series of 7 hands-on modules, our curriculum familiarizes students with these four topics.

Modules 1 and 2 provide an introduction to the actuator controller design. The mechanics of the brushless motors, their gear boxes and motor controllers, and the theory of simple control algorithms are taught via a lecture. Next, students construct a single leg of the robot mounted upside down on a metal plate to create what mimics a 3 degree of freedom robot arm. Using this arm and premade starter code, students program Bang-Bang and PD controllers and deploy them to the arm. Experimenting by changing parameters such as controller constants and loop rates, students gain real-world intuition on how these controllers work.

Modules 3 and 4 cover the derivation of the inverse kinematics for the 3 degree of freedom robot arm. This arm eventually becomes one of the 4 legs of the quadruped, so these kinematics are central to the locomotion of the robot. The modules culminate in implementing the kinematics in simulation and transferring to the real robot arm, introducing students to the Sim2Real pipeline.

Module 5 is dedicated to introducing students to the concept of Reinforcement Learning and its applications in robotics. A premade codebase allows for students to train a policy in simulation and then deploy the policy onto the physical robot.

Module 6, titled ”Pupper Assembly”, guides students through the entire assembly of the robot. This includes wiring, construction of the 3D-printed legs and body, setting the IDs of each of the 12 actuators, flashing the controller to the Teensy microcontroller, and binding the hand-held transmitter to the on-board receiver.

E. Course Outcomes

Thus far, our curriculum has been taught as independent study at Washington University in St. Louis, Foothill college, and at Stanford 4 times. The class enrollment at Stanford was 9, 17, 4, 15 for each of the four quarters the curriculum was taught. Students were undergraduates at all levels and had majors in mechanical engineering, computer science, electrical engineering, and mathematics. Students gave unanimously positive feedback. In a survey to students in the fourth offering of the course, students unanimously cited that they were more likely to continue in the fields of AI and robotics after taking the class than before.

The curriculum was advertised as introductory in AI and robotics, and most students (particularly first-years) had no experience in AI or robotics before taking the course. However, students were still able to develop impressive final projects using SoTA robot learning techniques. In fall of 2021, a group of 3 students developed a simple implementation of rapid-motor adaptation [citation] for Pupper and a group of 4 students used RL to endow Pupper with different locomotion styles based on a velocity command. In winter of 2023, 2 students added a vision setup to Pupper using Luxonis Oak-D Lite and gave Pupper the ability to autonomously track people. Also in winter of 2023, 2 students implemented [cite ChatGPT for Robotics] a simple method for using LLMs to task-plan for Pupper - endowing it with the ability to follow complex, nuanced text commands from users based on an API library developed in a previous final project. This final projects later became a task-planning lab completed by all students. Other student final projects included human pose-tracking, speech recognition, learning agility, and ball following for Pupper. These final projects showcase the diversity of AI methods that can be deployed on Pupper for students with limited experience.

IV. Discussion

Write after finishing rest

A. Conclusion

Write after finishing rest

Acknowledgment

References