2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)
Nancy, France. November 22-24, 2024

2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids) | 979-8-3503-7357-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/Humanoids58906.2024.10769865

STRIDE: An Open-Source, Low-Cost, and Versatile Bipedal Robot Platform for Research and Education

Yuhao Huang*, Yicheng Zeng*, and Xiaobin Xiong

Abstract— **In this paper, we present STRIDE, a Simple, Terrestrial, Reconfigurable, Intelligent, Dynamic, and Educational bipedal platform. STRIDE aims to propel bipedal robotics research and education by providing a cost-effective implementation with step-by-step instructions for building a bipedal robotic platform while providing flexible customizations via a modular and durable design. Moreover, a versatile terrain setup and a quantitative disturbance injection system are augmented to the robot platform to replicate natural terrains and push forces that can be used to evaluate legged locomotion in practical and adversarial scenarios. We demonstrate the functionalities of this platform by realizing an adaptive step-to-step dynamics based walking controller to achieve dynamic walking. Our work with the open-soured implementation shows that STRIDE is a highly versatile and durable platform that can be used in research and education to evaluate locomotion algorithms, mechanical designs, and robust and adaptative controls.**

Project Repository: https://github.com/well-robotics/STRIDE
Project Video: https://youtu.be/wJkxvUG6msU

I. Introduction

Bipedal robots are garnering unprecedented attention due to their potential capability to work in human society. Their potential to navigate in human environments and utilize human-sized tools have demonstrated significant value in alleviating people from completing repetitive dull, dear, dirty, and dangerous tasks. Existing platforms such as Boston Dynamics Atlas [1], Agility Robotics Digit [2], and Disney Research BD-1 [3] have already shown their capability in the manufacturing, transportation, and entertainment industries.

However, present-day, normal commercial [1]–[3] and research bipedal robot platforms [4], [5] are closed-source, mechanically un-configurable, and expensive. For instance, the previously purchasable robot Digit was priced at $250,000, and the cheaper robot Unitree G1 Edu is priced at over $30,000 in 2024, making them inaccessible to use in classrooms or even in most research labs, especially in third-world countries. Moreover, the increased complexity of the system reduces its durability; all these robots require a team of engineers for maintenance in research and development. Design modification and customization on commercial robots are also challenging. Consequently, researchers are prone to choose disparate platforms for their bipedal locomotion research [6]–[8], which challenge scientific comparisons and systematic evaluations of algorithms on similar robots. These issues can be partly addressed by increasingly realistic simulators. However, simulators also have limitations, particularly in areas such as contact simulation with deformable terrains [9]. All these factors hinder the researchers and educators from expediting further contributions to this field. Hence, the need for a unified platform that enables algorithm prototyping and evaluation is critical for fostering scientific progress in bipedal locomotion research.

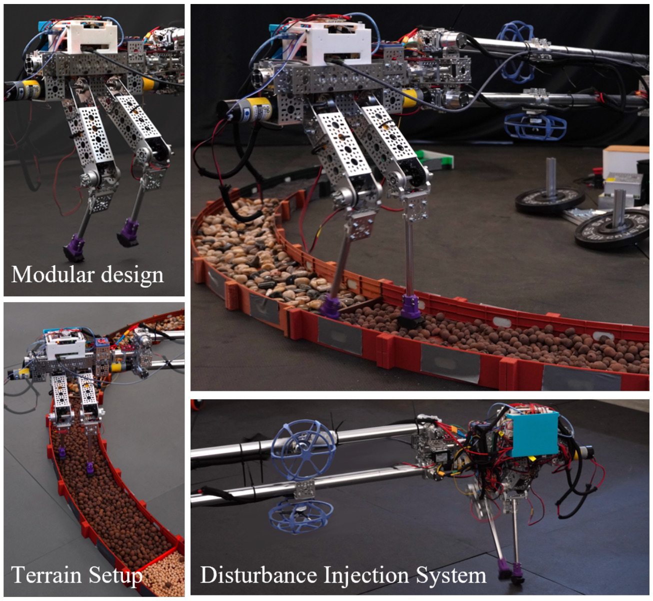

Fig. 1: Open-source robot STRIDE.

Some open-sourced bipedal robot platforms [10]–[14] have been developed to address these problems. However, they still encounter challenges in certain aspects. Some of these platforms heavily rely on existing commercial products, limiting access of users to detailed design information and hardware resources; the platforms based on custom-designed metal parts [10], [15] can require laborious and expensive machining, making them difficult to build and customize. Although these platforms are cheaper than existing commercial ones, they still represent a significant financial burden for educators and individual researchers. An increasing number of open-source platforms are opting for 3D-printed mechanical structures [11], [12], [14], but they often struggle with mechanical durability, particularly when performing locomotion with repeated impact with the ground. These structures also lack mechanical precision, making it difficult to evaluate the performance of the controllers. Therefore, there is still a huge necessity for having low-cost, customizable, and robust

* The authors contribute equally to this work.

[1] This work is supported in part by the University of Wisconsin-Madison Office of the Vice Chancellor for Research and Graduate Education.

The authors are with the Wisconsin Expeditious Legged Locomotion (WELL-Lab) at the University of Wisconsin-Madison. Corresponding to xiaobin.xiong@wisc.edu

bipedal robotic platforms in academia.

In this paper, we propose STRIDE, a planar bipedal robot platform to fulfill this urgent demand for accessible research and educational hardware platforms. The platform is low-cost (below $2000 in 2024) and easy to customize without the need of machining. The platform provides dynamic locomotion capabilities, and particularly, it features the following novel functionalities for evaluating legged locomotion:

1) Versatile Terrain Setup: A modular terrain setup is provided to enable the evaluation of locomotion over natural terrains.
2) Quantitative Disturbance Injection: A set of propellers is utilized to provide measurable push forces to the robot for robustness evaluation of the controllers.
3) Modular Components for Design Evaluation: A robot design with modular parts allows easy reconfigurations to test design optimization algorithms.

The whole design with software implementation is open-sourced at [16]. The video demonstration of STRIDE can be seen in [17].

The rest of this paper is organized as follows. Section II introduces the overall robot design. Section III describes an example of the control implementation for the robot. Section IV presents several experiments to demonstrate the performance, robustness, and versatilities of the platform. Finally, Section V draws a conclusion to our work and points out future directions.

II. System Design

Overview: The design of STRIDE aims to be low-cost, reconfigurable. We also exert effort to lower the assembly difficulty, making it straightforward even for novices. The mechanical design primarily consists of off-the-shelf metal components from goBILDA [18] and a small number of custom-designed 3D-printed parts, ensuring both strength and cost-efficiency. Its electrical components include widely accessible units such as Arduino and Raspberry Pi, further enhancing the affordability and accessibility of the robot. Its software consists of commonly used tools such as ROS2 [19], MuJoCo [20], and FROST [21], making the system modular and expandable.

Education Potentials: A wide range of robotic knowledge can be gained from building and implementing controllers on STRIDE, making it a versatile platform for teaching various topics, including but not limited to robot design, control theory and engineering, and ROS2. Specialized topics in legged robotics, such as floating-based system modeling, underactuated robotics theory, and trajectory planning and optimal control, can also be instrumented on this system. The open-source software repository serves as a template for implementing software on legged robots, further enhancing its educational value.

Research Potentials: Apart from the basic functionality of the robot, STRIDE is also designed with significant research potential. The control software is highly generalizable, allowing for the implementation, evaluation and comparisons of different control [22]–[27] and learning-based algorithms [28]–[30] for synthesizing bipedal walking. The modular environment setup can simulate outdoor conditions such as wind disturbances and rough terrain. Additionally, the modular design of the robot allows research evaluation of hardware design, such as robot design optimization. This can be conveniently implemented by using modular metal and 3D-printed components to adjust both the geometric and dynamic properties of linkages.

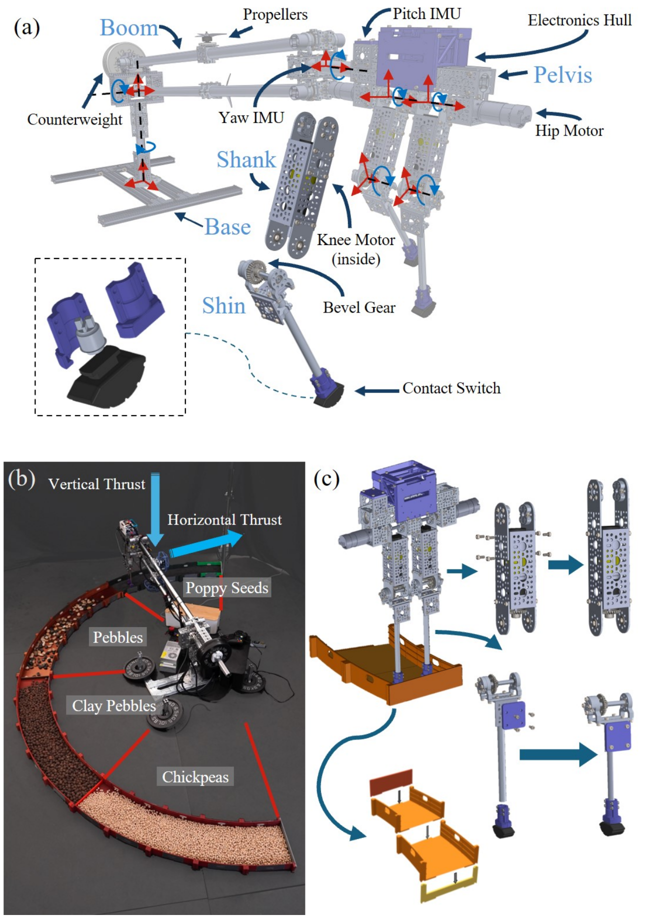

Fig. 2: (a) Mechanical design of STRIDE. (b) System setup with rough terrains and disturbance injection system. (c) Demonstration of modularity and easy customization.

A. Mechanical Components

Fig. 2 (a) shows the overall breakdown of the mechanical design of the robot. The system consists of a planar five-link bipedal robot and a parallel four-bar linkage boom that is connected to a fixed base. This configuration resembles the classical planar bipedal platforms such as RABBIT [5], AMBER [4], and ERNIE [31]. To ensure the robot has predominantly planar motion, the four-bar linkage boom constrains the robot in its sagittal plane and prevents lateral motion. Counterweights on the opposite end of the boom balance its gravitational forces, thereby minimizing the impact of boom dynamics to the robot dynamics. Two propellers are mounted on the four-bar linkages orthogonally as shown in Fig. 2 (b), which can provide accurate and measurable disturbance forces to the floating-base of the robot. A low-cost and versatile terrain setup is built for evaluating locomotion over natural terrains. All the components are modular and self-contained so that one can easily repurpose them for different purposes.

Robot Design: Each leg of the robot has two actuated joints: the hip motor is placed near the torso, and the knee motor is placed inside the thigh. Bevel gears are employed to transmit power from knee motors to the knee joints. This configuration reduces the inertia of the legs and enables rapid control of leg swinging. The brushed DC actuators are selected to meet both torque and angular speed criteria while maintaining a minimal gear ratio to reduce reflected inertia. All the actuators are from goBILDA (84 RPM, 93.6 kg·cm); it offers actuators with the same form factor but with different gear ratios and output shafts, which allows custom selections of actuators if needed. Contact detection is achieved through cheap contact switches with 3D-printed curved feet that can tolerate repetitive impacts with the ground.

Disturbance Injection: The propellers offer a more precise and measurable approach to inject disturbances to the robot, compared to the commonly seen method where a human operator simply pushes with unknown magnitudes [32]. The two propellers are installed orthogonally to provide disturbance forces to any direction in the plane, which allows thorough evaluations of the robustness of controllers.

Terrain Setup: The entire terrain setup consists of 3D-printed modular Lego-like structures that can be stacked up piece by piece. It can contain rough materials such as poppy seeds, garbanzo beans, cobblestones, and rocks to emulate natural terrains. In this design, we choose poppy seeds, pebbles, clay pebbles, and chickpeas as the filling materials. Poppy seeds are granular and soft, while pebbles are relatively large and slippery. Clay pebbles are of the same size as pebbles but with rougher surfaces, whereas chickpeas have similar roughness with smaller diameters. The cost of the terrain is estimated at $200 for one full circular track.

B. Electronic Components

The electronic system is designed to be modular and easily reconfigurable for different research purposes. Fig. 3 (a) depicts the electrical components and communication protocols used in the robot. The whole electrical system has a user computer (Raspberry Pi 4b+) for implementing robot control and sensor processing and a low-level microcontroller (Arduino Mega 2560 R3) that has implemented motor control algorithm. Communication between the Raspberry Pi and the Arduino is implemented through a USB. These components are chosen based on the balance of cost and performance, ensuring that the robot has sufficient computation capacity. Additionally, all electronic components are mounted on different layers of a housing unit, allowing for easy inspection and substitution of individual components. This design simplifies hands-on assembly and eases the debugging of the electronic system.

Two IMUs (SparkFun ICM-20948 9DoF IMU) and two contact switches are connected to the Raspberry Pi for gait synthesis. One IMU is placed on the robot pelvis to measure the pelvis angle and angular velocity, while another is mounted at the end of the four-bar linkage, as shown in Fig. 2 (a), to measure the rotational velocity of the circular motion. Compared to an encoder, an IMU is easier to assemble while providing sufficiently accurate measurements. Two contact switches are placed on the foot to measure the foot-ground contact. The Arduino Mega controls the brushed DC actuators by sending Pulse Width Modulation (PWM) signals to the Motor Drivers (VNH5019 12A, 5.5-24V); the Arduino also reads the pulse counts from the encoders (1993.6 pulses per revolution) built inside the actuators. The motor drivers, Arduino, and Raspberry Pi are powered by a 12V power supply (MEISHILE 12V 40A 480W).

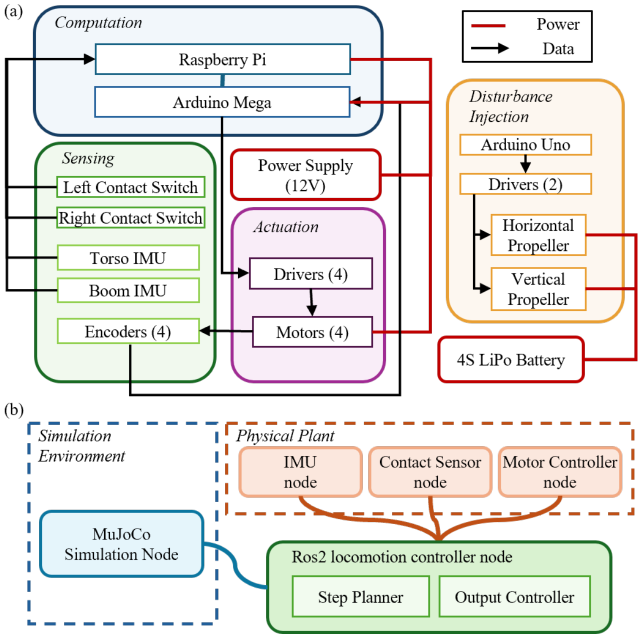

Fig. 3: (a) Electrical components and wiring topology. (b) Software architecture, in which the ROS2 controller node can be customized by the user.

The disturbance injection system includes two propellers, two brushless DC motors (T-Motor V2207 2550KV), and two motor controllers (Tiger Motor F35A) connected to an Arduino Uno. The Arduino receives commands from a joystick via a USB communication, and it sends out control signals (DShot300) to the motor driver to control the motors. This setup allows users to provide both continuously changing and fixed amount disturbance forces to the robot based on their needs.

The entire electronic setup is easily reconfigurable. The disturbance injection system can share the same Arduino as the motor control system, eliminating the need for an additional Arduino Uno. Additionally, the joystick can be connected to the Raspberry Pi, allowing it to control both the robot and the disturbance injection system if needed.

C. Software

The software architecture of the robot is designed to be efficient, hardware-friendly, modular, and expandable. The code is developed using state-of-the-art software environment ROS2, making it suitable for education. Hardware interfaces and controllers are implemented in `C++` and leverage the `Eigen` library for linear algebras, ensuring efficient computation and compatibility with various hardware configurations. Along with these codes, a separate software package for simulation and control is provided in MATLAB, allowing users to quickly validate their research works and visualize results through straightforward simulation and animation.

Fig. 3 (b) depicts the overall architecture of our software package that has two major components: hardware interfaces and the controller. The hardware interfaces process raw sensor readings to publish them as appropriate message types for walking control. The controller subscribes to sensor messages, plans desired motion, and solves for the motor control commands which are published as ROS2 topics. An example of the controller implementation of walking is described in section IV. To support custom implementations of controllers, we also provide all the kinematics and dynamics libraries of the robot in the software package.

In parallel to the hardware nodes, a simulation node using MuJoCo [20] has been implemented. In this simulation, the robot is constrained in its sagittal plane virtually to mimic the designed planar motion. Sensor data in the simulation are formatted and directly sent to the controller node. In this way, the same controller node can be evaluated in both simulation and hardware. This architecture minimizes the sim-to-real transfer iterations and ensures rapid implementation debugging in simulation before deployment on the hardware, which minimizes the potential risks of damaging the hardware with buggy controls. It also facilitates debugging in various scenarios by replicating real-world conditions, e.g., observing the effects of excessive or insufficient feedforward torque, the impact of sensor noise, and variation in locomotion performance w.r.t. controller parameters.

D. *Modularity and Customization*

The robot is designed to facilitate the customization of both hardware and software. goBILDA provides metal parts of various lengths and actuators with different gear ratios. Additionally, the mass and inertia properties of the robot can be easily modified by attaching 3D-printed parts to its body. The design approach simplifies modifications on both the kinematic and dynamic properties of the robot. An illustration of its modularity is illustrated in Fig. 2 (c). To change the link lengths or add masses to the links, one only needs to remove a few screws, swap the parts, and tighten the screws again. The terrain setup is designed for easy assembly and disassembly, with flexible adjustments in its length and the terrain it contains. As shown in Fig. 2 (c), terrain blocks can be easily stacked or removed. A connector block can be inserted to combine two terrain blocks while separating different terrain materials if needed. Moreover, the electrical components can be readily substituted for any appropriate ones due to the modular design of the electronic hull. The main computer can be swapped to other single-board computers such as an Intel UpBoard or NUC, and the motor controllers can be upgraded to advanced motor drivers that are capable of performing current regulation.

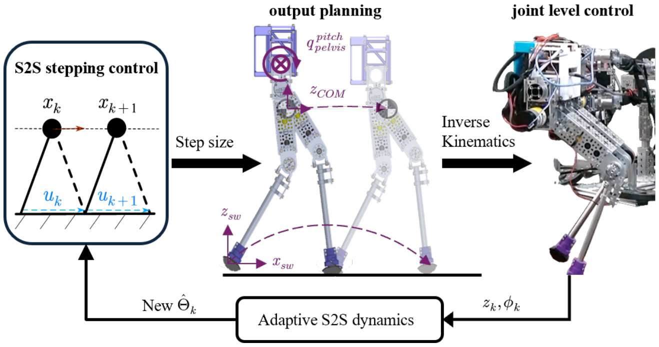

Fig. 4: Structure of the walking controller Implementation based on S2S dynamics.

The software, both the simulation and hardware, are written as ROS2 packages, and thus can be easily customized for different educational and research purposes. `C++` enables highly efficient computation and direct integration with ROS2, while ROS2 allows for seamless integration with other open-source packages in terms of control, planning, and estimation. For example, a quadratic programming (QP) based controller from [32] can be easily integrated into the control implementation example, by substituting the Inverse Kinematics (IK) and PD-based controller. The modular design of both the software and hardware will facilitate the rapid evaluation of novel controls and hardware designs.

III. Control Implementation example

This section describes a detailed control implementation example of STRIDE. The Step-to-Step Dynamics (S2S) based walking controller [32], [33] is used to synthesize the desired walking behaviors due to its general formulation of walking that can be used for robots with different hardware configurations, its robustness and adaptation to disturbances, and its straightforward implementation.

A. *Hybrid Linear Inverted Pendulum Model*

Bipedal walking can be naturally described by the step-to-step (S2S) dynamics [34]. Mathematically, the S2S dynamics can be treated as a Poincaré return map at the Poincaré section which is normally chosen at the surface of the ground impact. The state of the robot is chosen as $x = [q, \dot{q}]^T$ where q represents the generalized coordinates of the robot. Let x_i^- denote the pre-impact state of the i-th ground impact during walking. The S2S dynamics of the robot can be written as $x_{k+1}^- = \mathcal{P}(x_k^-, \tau(t))$ where $\tau(t)$ is the control input over time.

Due to the highly nonlinear nature of the S2S dynamics of real bipedal robots, it is hard to acquire the analytical form of their S2S dynamics. Therefore, a linear approximation to the robot S2S dynamics is proposed in [32] which utilizes a reduced-order model called Hybrid Linear Inverted Pendulum (H-LIP). The H-LIP is a model with a constant center of mass (COM) height and two legs with point feet. Based on the number of feet in contact with the ground, the system is either in a single-stance phase (SSP) or a double-stance phase (DSP). The SSP phase is the same as the passive Linear Inverted Pendulum (LIP) model, and the DSP phase is assumed to have a constant horizontal COM velocity. Chosen the state of the H-LIP model as $x = [p, v]^T \in \mathbb{R}^2$, where p and v are the horizontal COM position and velocity w.r.t. the stance foot, respectively, the S2S dynamics of the H-LIP model then can be written as:

$$x_{k+1}^- = A^H x_k^- + B^H u_k, \tag{1}$$

where $u_k \in \mathbb{R}$ represents the step size of the H-LIP model. The detailed derivation of the system matrix and input matrix can be found in [32]. The robot reduced-order S2S dynamics can be written as:

$$x_{k+1}^- = A^H x_k^- + B^H u_k + w_m, \tag{2}$$

where $w_m = \mathcal{P}(x_k^-, \tau(t)) - A^H x_k^- - B^H u_k$ represents the integrated residual error between the S2S dynamics of the H-LIP model and the S2S dynamics of the real robot COM during one step. By constructing periodic walking gaits, w_m lies in a bounded set $w_m \in W$ [32]. Then a state feedback controller called H-LIP based stepping controller can be designed to mitigate the error between the robot COM state and H-LIP state: $\mathbf{e} = x^R - x^H$. Taking $u^R = u_k^H + K(x_k^R - x_k^H)$ yields the error S2S dynamics:

$$\mathbf{e}_{k+1} = (A + BK)\mathbf{e}_k + w_k. \tag{3}$$

The feedback K can be designed through LQR or deadbeat controller to stabilize the system with $eig(A + BK) < 1$.

The period orbits of the S2S dynamics of the robot can be characterized based on the number of steps completed during a walking period. Specifically Period-1 (P1) and Period-2 (P2) orbits are investigated.

P1 Orbit: The desired step size is uniquely determined by $u_d = v_d T$ where v_d is the given desired walking velocity and T is the desired step duration. Then the desired pre-impact state can be uniquely determined through the SSP dynamics of the H-LIP model and the desired step size.

P2 Orbit: Different from the P1 orbit, the P2 orbit that can realize a desired velocity v^d is not unique. The desired step sizes satisfy $u_L + u_R = 2v^d T$ where subscripts $_{L/R}$ denote the stance foot of that step, and T is the step duration. In order to uniquely realize a P2 orbit, one should first choose a step size for the right stance or left stance foot. The desired states can then be uniquely determined for each step.

B. Adaptive Step-to-step Dynamics

The H-LIP based model controller provides a baseline where the error of state tracking can be stabilized and bounded. To achieve higher accuracy and better performance, following the formulation in [33], we leverage a static parametric model to reduce this modeling error; we then apply an adaptive controller to achieve better velocity tracking. To achieve this goal, we first review the robot model in the form of an adaptive control scheme. Generally, a static parametric model [35] is:

$$z_k = \Theta^{*T} \phi_k, \tag{4}$$

where z_k and ϕ_k are measurable signals, and Θ^* is the model we are trying to adapt to. According to the S2S dynamics of the H-LIP model developed above, we can expect a linear dynamics model with a constant offset:

$$\underbrace{x_k}_{z_k} = \underbrace{(A \quad B \quad C)}_{\Theta^{*T}} \underbrace{(x_{k-1} \quad u_{k-1} \quad 1)^T}_{\phi_k}, \tag{5}$$

where C represents the constant offset due to the nonlinear robot dynamics during stepping. Giving z_k and ϕ_k at each step, we can use the update law:

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} + \Gamma \phi_k (\phi_k^T \phi_k)^{-1} (z_k - \hat{\Theta}_{k-1}^T \phi_k^T), \tag{6}$$

where Γ is a tunable gain for controlling the update size. The performance of the adaptive controller will be evaluated in the experiment section to emphasize the capability of prototyping advanced control algorithms on STRIDE.

C. Gait Design and Stepping on STRIDE

To apply the aforementioned H-LIP based approach on STRIDE, the designed gait should fulfill the requirement of the H-LIP model: the vertical COM position z_{com} should be approximately constant w.r.t. the stance foot, the vertical position of the swing foot will be periodically lift-off and strike the ground, and the horizontal position of the swing foot will achieve the desired step size u^d from the H-LIP based stepping controller. To fully constrain the walking, the orientation of the pelvis q_{pelvis}^p is always controlled to certain desired values. Therefore the output during SSP is:

$$\mathcal{Y} = \begin{bmatrix} z_{COM} \\ x_{sw} \\ z_{sw} \\ q_{pelvis}^p \end{bmatrix} - \begin{bmatrix} z_{COM}^d \\ x_{sw}^d \\ z_{sw}^d \\ q_{pelvis}^{p^d} \end{bmatrix}. \tag{7}$$

Since there are no compliant components in the mechanical design, and under the assumption of perfect plastic ground contact [36], we assume only SSP will occur during walking. Therefore, only the trajectories of the SSP are designed.

The desired pitch angle of the pelvis is chosen to be constant. The rest of the output trajectories are designed with Bézier polynomials. The desired horizontal trajectory of the swing foot is designed as:

$$x_{sw}^d = (1 - b_h(t)) x_{sw}^+ + b_h(t) u_x^d, \tag{8}$$

where x_{sw}^+ is the horizontal position of the swing foot w.r.t. the stance foot at the beginning of the current SSP, u_x^d is the desired step size from the S2S based stepping controller, and $b_h(t)$ is a Bézier polynomial that transits from 0 ($t = 0$) to 1 ($t = T_{ssp}$). The time variable t will reset when the robot swaps its support leg. The desired vertical COM position is controlled to a constant z_0 which is the constant height of H-LIP. Due to a small jump when swapping the support leg, the desired trajectory should achieve a small transit from COM height after impact to the constant value. The desired trajectory is constructed as:

$$z_{COM}^d = (1 - b_h(t)) z_{COM}^+ + b_h(t) z_0, \tag{9}$$

where z_{COM}^{+} is the vertical position of the swing foot w.r.t the stance at the beginning of the current SSP, and $b_h(t)$ is the same Bézier polynomial aforementioned. The vertical position of the swing foot is controlled to perform periodic lift-off and touch-down behaviors. Another Bézier polynomial b_v is used to design the trajectory in which the vertical position of the swing foot will first transit from 0 $(t = 0)$ to z_{sw}^{max} $(t = \frac{T_{SSP}}{2})$ and then go back to z_{sw}^{neg} $(t = T_{SSP})$. z_{sw}^{max} is a user-specified constant that determines the foot-ground clearance, and z_{sw}^{neg} is a small negative number for making sure the foot will strike the ground at the end of SSP phase. The designed z_{ws}^{d} is designed as:

$$z_{sw}^{d}(t) = b_v(t, z_{sw}^{max}, z_{sw}^{neg}). \quad (10)$$

D. Joint-level Controller Design

Newton-Raphson inverse kinematics is employed to solve the joint trajectories from the desired output trajectories. The joint trajectories are then tracked by a PD plus feedforward controller. The feedforward torque is calculated via gravity compensation, and the desired torque is mapped to voltages that is sent to the motor drivers via PWM at 20kHz.

Gravity Compensation: The Euler-Lagrange dynamics of STRIDE during SSP is:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = Bu + J_s^T F_s, \quad (11)$$

where $M(q)$ is the mass matrix, $C(q, \dot{q})$ is the Coriolis and centrifugal forces, $G(q)$ is the gravity vector, B is the input matrix, J_s is the Jacobian matrix of the stance foot, u is the control input and F_s is the ground reaction force of that foot. Taking $\ddot{q} = \dot{q} = 0$, the proposed gravity compensation method can be formulated as a least-square optimization:

$$\min_{u} \quad ||G(q) - Bu - J_s^T F_s||^2, \quad (12)$$

which is solved using linear algebra functions in `Eigen`.

Motor Control: Once the torques are solved, they are converted to a PWM value for motor control. The model of brushed DC motor with constant armature inductance described in [37] is written as:

$$V = k_e N \omega + IR, \; \frac{\tau_m}{N} = k_t I, \quad (13)$$

where V, I are the voltage supplied and current through the motor armatures, R is the resistance of the armatures, k_e and k_t are the back EMF constant and motor torque constant, respectively, N is the gear ratio, τ_m is the torque at output shaft and ω is the angular velocity of the motor at the output shaft. The torque at output shaft can then be mapped to the voltage by solving for V:

$$V = k_e N \omega + \frac{R}{k_t N}\tau_m. \quad (14)$$

IV. System Demonstrations

We then evaluate the performance of the proposed platform through several hardware experiments, with the controller developed in the previous section. Four experiments have been designed to analyze the potential of the platform: as an educational platform, as a testing platform for evaluating locomotion controls, and as a testing platform for design optimization. Fig. 2 (b) has depicted the experimental setup, and the experiment results can be seen in [17].

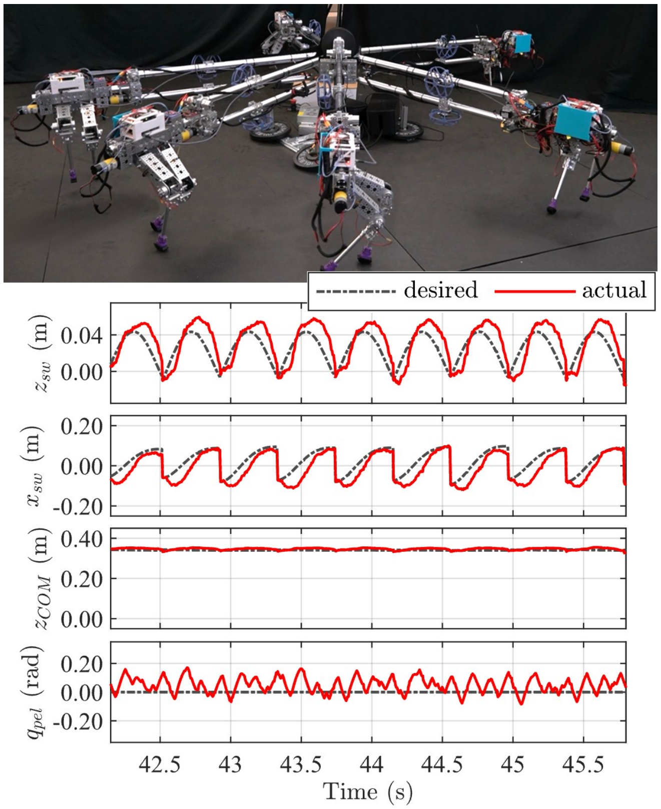

Fig. 5: Stable walking with output tracking results.

A. Rigid Ground Walking

This experiment aims to demonstrate the capability of using the platform as an educational platform for legged locomotion. An H-LIP based S2S controller without adaptation is employed to synthesize desired walking behaviors. The walking height is chosen as $z_d = 0.34m$, and the desired walking speed is chosen as $v_d = 0.2m/s$. The desired walking periodic orbit is synthesized as a P1 orbit, with the walking period specified at $T = 0.3s$. Fig. 5 illustrates the output trajectories of the robot.

Given that the hardware designs, electronic components, software, and walking controllers are open-sourced, students can reproduce the platform and the experiment following the provided detailed guidance. Learning outcomes can be analyzed based on the success of realizing walking behaviors. Moreover, the quality of the implementation can be evaluated based on the tracking errors between the desired and actual outputs. The errors currently come from two primary sources: (1) inverse kinematics error when mapping the desired outputs to desired joint outputs, which are relatively small, and (2) joint tracking errors, which are generally more apparent. A smaller error indicates better performance in the mechanical assembly, sensor filtering, and control implementation, reflecting the learning outcome of the students.

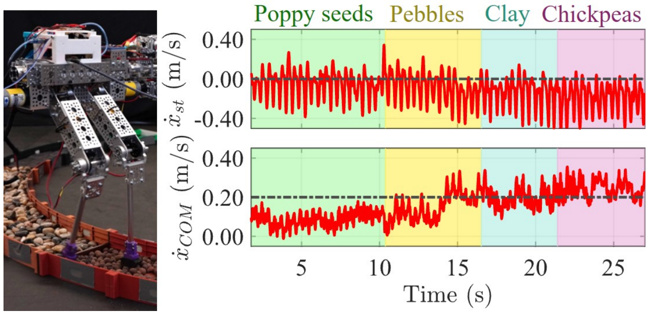

Fig. 6: Rough terrain walking experiment.

B. Walking over Modular Challenging Soft Terrain

This experiment explores the potential of the platform for testing the robustness of different walking algorithms over rough terrains. Different terrains are used to simulate natural environments, making it easier to test real-world mobility indoors. The rough terrains are composed of poppy seeds, pebbles, clay pebbles, and chickpeas, respectively; they can be swapped to any other terrains easily. The walking parameters for this experiment are identical to those in the rigid ground walking experiment for comparative purposes.

The stance foot and COM velocities in the horizontal direction are chosen as performance indicators. Fig. 6 has depicted the walking performance of the robot on different terrains. Due to the varying surface roughness and hardness of the materials, the tracking behaviors of COM velocity and stance foot velocity of the robot differ. COM horizontal velocity of the robot is slower than expected when walking on poppy seeds and more jittery when walking on chickpeas and clay pebbles. The stance foot horizontal velocity of the robot is more slippery when walking on chickpeas and clay pebbles compared with walking on poppy seeds. The noticeable differences in velocity tracking across various terrains validate the effectiveness of the system in testing the robustness of walking algorithms on different surfaces.

C. Quantitative Disturbance Injection

This experiment aims to provide the potential of applying the platform for advanced adaptive control validation and comparison. The walking parameter is chosen as the same as the rigid ground walking experiment, and the adaptive gain is chosen as $\Gamma = 0.4\mathtt{I}$ where $\mathtt{I}$ is the identity matrix. The horizontally mounted propeller provides a measurable disturbance of 3N. Fig. 7 has depicted the step tracking performance. Before the horizontal disturbance is applied, both controllers with or without adaptation have stable walking tracking. After the disturbance force is introduced, it can be easily seen that the adaptive controller provides better velocity tracking in terms of step sizes after a certain time of adaptation. The walking is stabilized to a mean step size of $0.08m$. A clear performance difference between the controller with and without adaptation has validated the use of this platform as a testing platform for the evaluation of advanced controls.

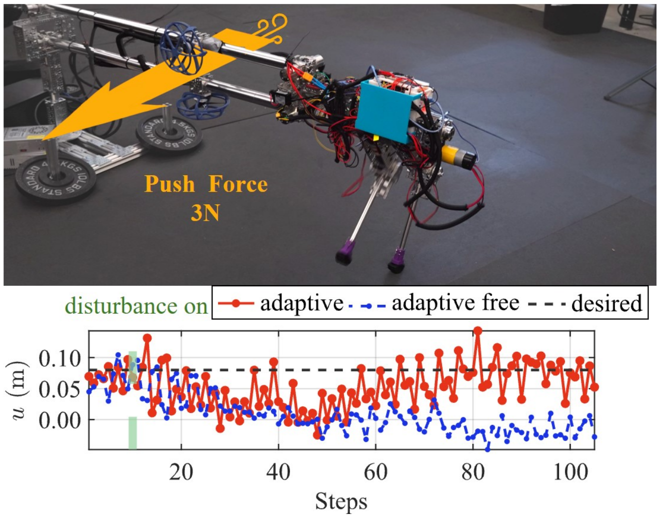

Fig. 7: Adaptive control with disturbance experiment result.

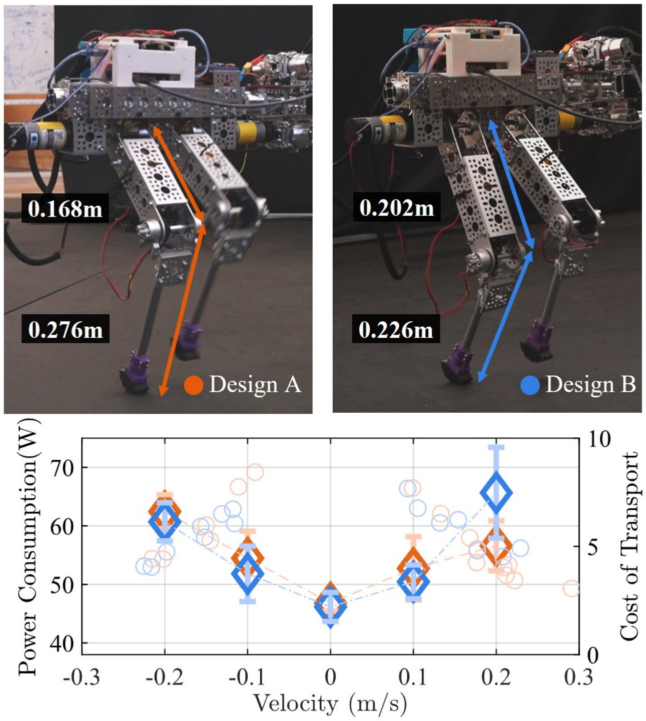

Fig. 8: Power Consumption (diamond markers) and Cost of Transport comparison (circles) on two different leg designs.

D. Design Evaluation Experiment

In this experiment, we explore a potential scenario where design optimization is implemented to determine the optimal thigh-to-shin ratio. We implement two design candidates by changing the goBILDA parts shown in Fig. 2 (c). We then evaluate the two designs by comparing the power consumption and the cost of transport (COT) of walking, as shown in Fig. 8. The power consumption is lowest when the robot steps in place. Design A has relatively lower power consumption at higher positive velocities, while design B consumes less power at negative velocities. The COT for both designs is similar, with larger COT values occurring when the speed is smaller. As the speed increases, the COT decreases. This validates the purpose of this platform for rapid design evaluations on real hardware.

V. Conclusion and Discussion

In conclusion, we present an open-source bipedal robotic platform that is low-cost, versatile, modular, and customizable. The hardware system requires no machining, and it is primarily built from modular off-the-shelf parts for easy customization. The implemented controller using the S2S framework demonstrates the capability of deploying state-of-the-art walking controllers on the platform to locomote over natural terrains with push disturbances. Experiments validate its value for future education and research activities.

In the future, we plan to use STRIDE to evaluate novel algorithms in motion planning, state estimation, feedback control, reinforcement learning, and bipedal locomotion to further demonstrate its value in research. In education, we plan to introduce STRIDE as a platform of choice for the course ME601: Advanced Robotics at UW-Madison, educating students on theories and engineering. We envision that STRIDE will promote legged research and cultivate students with various expertise required to develop the next generations of dynamic, robust, and efficient legged robots.

References

[1] Atlas: https://bostondynamics.com/atlas/.

[2] Aiglity Robotics Digit: https://agilityrobotics.com/products/digit.

[3] Disney Research BD-1: https://spectrum.ieee.org/disney-robot.

[4] E. Ambrose, W.-L. Ma, C. Hubicki, and A. D. Ames, "Toward benchmarking locomotion economy across design configurations on the modular robot: Amber-3m," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 1270–1276.

[5] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. Canudas-De-Wit, and J. Grizzle, "Rabbit: a testbed for advanced control theory," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57–79, 2003.

[6] J. Reher and A. D. Ames, "Dynamic walking: Toward agile and efficient bipedal robots," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. Volume 4, 2021, pp. 535–572, 2021.

[7] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. D. Prete, "Optimization-based control for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 40, pp. 43–63, 2024.

[8] S. Katayama, M. Murooka, and Y. Tazaki, "Model predictive control of legged and humanoid robots: models and algorithms," *Advanced Robotics*, vol. 37, pp. 1–18, 02 2023.

[9] C. K. Liu and D. Negrut, "The role of physics-based simulators in robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. Volume 4, 2021, pp. 35–58, 2021.

[10] J. Li, J. Ma, O. Kolt, M. Shah, and Q. Nguyen, "Dynamic loco-manipulation on hector: Humanoid for enhanced control and open-source research," *ArXiv*, vol. abs/2312.11868, 2023.

[11] F. Grimminger, A. Meduri, M. Khadiv, J. Viereck, M. Wüthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, T. Flayols, J. Fiene, A. Badri-Spröwitz, and L. Righetti, "An open torque-controlled modular robot architecture for legged locomotion research," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3650–3657, 2020.

[12] M. Lapeyre, P. Rouanet, J. Grizou, S. Nguyen, F. Depraetre, A. Falher, and P.-Y. Oudeyer, "Poppy project: Open-source fabrication of 3d printed humanoid robot for science, education and art," 09 2014.

[13] J. Ramos, Y. Ding, Y.-W. Sim, K. Murphy, and D. Block, "Hoppy: An open-source kit for education with dynamic legged robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4312–4318.

[14] P. Allgeuer, H. Farazi, M. Schreiber, and S. Behnke, "Child-sized 3d printed igus humanoid open platform," *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 33–40, 2015.

[15] A. SaLoutos, E. Stanger-Joncs, Y. Ding, M. Chignoli, and S. Kim, "Design and development of the mit humanoid: A dynamic and robust research platform," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023, pp. 1–8.

[16] Github: github.com/well-robotics/STRIDE.

[17] Experiement: https://youtu.be/wJkxvUG6msU.

[18] goBILDA: https://www.gobilda.com/structure.

[19] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[20] 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*, 2012, pp. 5026–5033.

[21] A. Hereid and A. D. Ames, "Frost: Fast robot optimization and simulation toolkit," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vancouver, BC, Canada: IEEE/RSJ, Sep. 2017.

[22] G. Gibson, O. Dosunmu-Ogunbi, Y. Gong, and J. Grizzle, "Terrain-aware foot placement for bipedal locomotion combining model predictive control, virtual constraints, and the alip," *arXiv preprint arXiv:2109.14862*, 2021.

[23] Z. He, J. Wu, J. Zhang, S. Zhang, Y. Shi, H. Liu, L. Sun, Y. Su, and X. Leng, "Cdm-mpc: An integrated dynamic planning and control framework for bipedal robots jumping," *IEEE Robotics and Automation Letters*, 2024.

[24] Y. Gong and J. W. Grizzle, "Zero dynamics, pendulum models, and angular momentum in feedback control of bipedal locomotion," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 12, p. 121006, 2022.

[25] X. Xiong, Y. Chen, and A. D. Ames, "Robust disturbance rejection for robotic bipedal walking: System-level-synthesis with step-to-step dynamics approximation," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 697–704.

[26] Z. Xiang, V. Paredes, and A. Hereid, "Adaptive step duration for precise foot placement: Achieving robust bipedal locomotion on terrains with restricted footholds," *arXiv preprint arXiv:2403.17136*, 2024.

[27] B. Acosta and M. Posa, "Bipedal walking on constrained footholds with mpc footstep control," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023, pp. 1–8.

[28] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control," *arXiv preprint arXiv:2401.16889*, 2024.

[29] G. A. Castillo, B. Weng, S. Yang, W. Zhang, and A. Hereid, "Template model inspired task space learning for robust bipedal locomotion," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 8582–8589.

[30] "Revisiting reward design and evaluation for robust humanoid standing and walking," *arXiv preprint arXiv:2404.19173*, 2024.

[31] T. Yang, E. R. Westervelt, J. P. Schmiedeler, and R. A. Bockbrader, "Design and control of a planar bipedal robot ERNIE with parallel knee compliance," *Auton. Robots*, vol. 25, no. 4, pp. 317–330, Nov. 2008.

[32] X. Xiong and A. Ames, "3-d underactuated bipedal walking via h-lip based gait synthesis and stepping stabilization," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2405–2425, 2022.

[33] M. Dai, X. Xiong, J. Lee, and A. D. Ames, "Data-driven adaptation for robust bipedal locomotion with step-to-step dynamics," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 8574–8581.

[34] D. Hobbelen and M. Wisse, *Limit Cycle Walking*, 06 2007.

[35] P. Ioannou, *Adaptive Control Tutorial (Advances in Design and Control)*. USA: Society for Industrial and Applied Mathematics, 2006.

[36] Y. Hurmuzlu and D. B. Marghitu, "Rigid body collisions of planar kinematic chains with multiple contact points," *The International Journal of Robotics Research*, vol. 13, no. 1, pp. 82–92, 1994.

[37] K. Lynch and F. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge Univeristy Press, 2017.