HOPPY: An open-source and low-cost kit for dynamic robotics education

Joao Ramos[1,2], Yanran Ding[1], Young-woo Sim[1], Kevin Murphy[1], and Daniel Block[2]

Abstract—This letter introduces HOPPY, an open-source, low-cost, robust, and modular kit for robotics education. The robot dynamically hops around a rotating gantry with a fixed base. The kit lowers the entry barrier for studying dynamic robots and legged locomotion in real systems. The kit bridges the theoretical content of fundamental robotic courses and real dynamic robots by facilitating and guiding the software and hardware integration. This letter describes the topics which can be studied using the kit, lists its components, discusses best practices for implementation, presents results from experiments with the simulator and the real system, and suggests further improvements. A simple controller is described to achieve velocities up to $2m/s$, navigate small objects, and mitigate external disturbances (kicks). HOPPY was utilized as the topic of a semester-long project for the Robot Dynamics and Control course at the University of Illinois at Urbana-Champaign. Students provided an overwhelmingly positive feedback from the hands-on activities during the course and the instructors will continue to improve the kit for upcoming semesters.

Index Terms—Legged Robots; Education Robotics; Engineering for Robotic Systems

Fig. 1. The hopping robot HOPPY for hands-on education in dynamic control and legged locomotion. Available at http://github.com/RoboDesignLab/HOPPY-Project.

I. INTRODUCTION

The imminent robotics revolution will employ robots as ubiquitous tools in our lives. Many machines are already being widely used in factories, assembly lines, and, more recently, in automated warehouses [5]. However, most of the tasks performed by these robots are quasi-static, which means that the robot can stop mid-motion without destabilizing (falling down). In contrast, humans can perform dynamic motions in order to complete tasks, like running or weightlifting, more efficiently. In this scenario, the task cannot be interrupted mid-motion. For instance, think how a runner cannot instantaneously freeze motion between steps without falling, or how a Olympic weightlifter cannot statically lift the payload above his/her head. Thus, we must train the next generation of roboticists to create machines that are capable of dynamically performing physical tasks like humans and other animals. Dynamic motions impose unique challenges related to mechanical robustness, actuation saturation, control rates, state estimation quality, and more. However, performing experiments with dynamic robot motions is challenging because capable hardware is expensive and not readily available, and errors can quickly lead to terminal hardware damage [6]. To address this issue, researchers created platforms focused on physical robustness, low-cost, and modularity [3], [7].

Authors are with the [1]Department of Mechanical Science and Engineering and the [2]Department of Electrical & Computer Engineering at the University of Illinois at Urbana-Champaign, USA. jlramos@illinois.edu
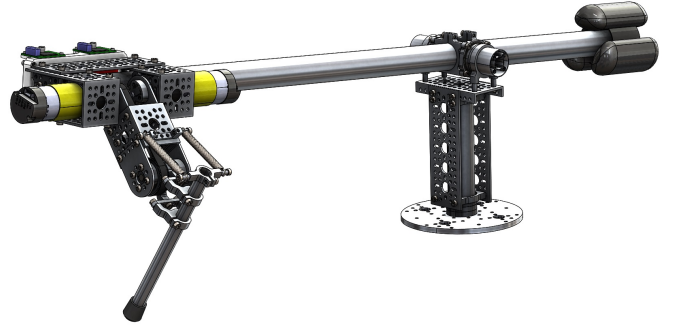
Although these dynamic platforms have a significant lower cost than most robots used for research, their use in large-scale education is still impractical. The cost associated with purchasing each unit and the requirement of fabrication tools (CNC machines or high-precision 3D printers), custom electronics, and low-level computing for control and simulation prohibit their use in large cross-disciplinary Engineering courses. In response, this letter introduces HOPPY, a low-cost kit designed for studying dynamic robots. The kit costs under $500 and is composed of the robot hardware, which can be build exclusively with off-the-shelf components, and a simulator based on the widely adopted software *MATLAB*. The kit covers most of the topics of fundamental robotics courses such as kinematics, dynamics, controls, trajectory generation, simulation, and more. As a tool for nurturing active-learning in robotics, the kit was implemented as a hands-on and semester-long project for the Spring 2020 *Robot Dynamics and Control (ME446)* course at the University of Illinois at Urbana-Champaign (UIUC) (http://youtu.be/6O6czr0YyC8). This hands-on approach to education is becoming increasingly popular due to its proven effect on enabling enduring, deeper, and more significant learning [11], [14]. With this manuscript, we make open-source the robot model, its MATLAB-based simulator, the bill of materials, and assembly instructions. Users can modify and adapt the hardware, software, and simulator as needed for their application and budget. This manuscript and the kit are intended for educators interested in implementing hands-on robotics activities, and for students, researchers, and hobbyists interested in experimentally learning about dynamic robots and legged locomotion.

This letter is organized as follows. First we describe the topics which are covered by the kit for implementation in
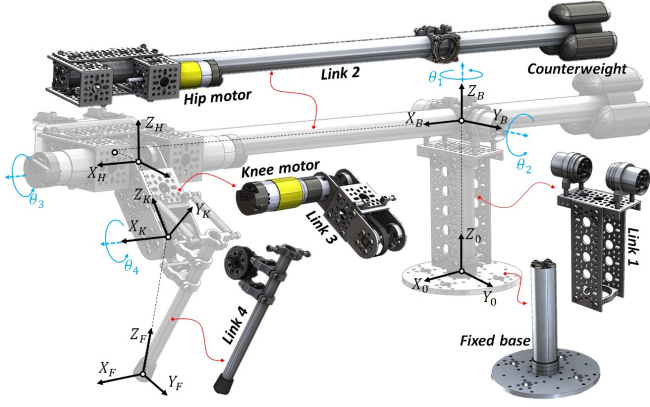
Fig. 2. Kinematic transformations and the five rigid-bodies which compose HOPPY. The first two joints (θ_1 and θ_2) are passive while joints three and four are driven by electric motors.

fundamental robotics courses. Next, we list the open-source files and instructions made available with this manuscript. We then suggest possible modifications or further improvements to adapt the hardware to the needs of particular educators or researchers. Finally, we discuss our experience implementing the kit in the ME446 course at UIUC and address future work.

II. ROBOTICS TOPICS COVERED

This section describes the (non-exhaustive) list of topics covered in the Robot Dynamics and Control course which are explored using the HOPPY kit. Although here we describe how the kit was modelled and implemented at UIUC, each topic may be explored with more or less detail according to the content of other courses.

When HOPPY hops around the gantry, it behaves as a *hybrid system* [19]. This means that the dynamics have continuous evolution when the robot is on the ground (stance phase) or airborne (flight phase), but it is interrupted by discrete events when the foot hits the ground (touchdown) or when it takes-off. In this section we describe the equations of motion during stance and flight phases, as well as the transitions between the two.

A. Kinematics

The kinematic transformations of HOPPY are shown in fig. 2. The robot consists of a two degrees-of-freedom (DoF) active leg attached to a supporting gantry with two additional DoF's. The origin coordinate frame $X_0 Y_0 Z_0$ at the base of the rotating gantry is fixed to the ground. Joints $\theta_{1,2}$ are passive (no actuation) while joints $\theta_{3,4}$ are driven by electric motors described in section II-C. When hopping, the robot can translate horizontally around θ_1 and vertically around θ_2. If the gantry length L_B is sufficiently long, the robot moves approximately in a 2D (Sagittal) plane [19]. The transformation between coordinate frames is described by the Homogeneous Transformation Matrices (HTM) given by T_{k+1}^k, which converts a vector p^{k+1} expressed in coordinate

frame $k+1$ into a vector p^k expressed in coordinate frame k [15]:

$$p^k = T_{k+1}^k p^{k+1} = \begin{bmatrix} R_{k+1}^k & d^k \\ 0_{1\times 3} & 1 \end{bmatrix} p^{k+1}. \quad (1)$$

Where the term $0_{1\times 3}$ represents a 1 by 3 matrix of zeros. The HTM is composed of a rotation matrix $R_{k+1}^k \in SO(3)$ and a Cartesian translation $d^k \in \mathbb{R}^3$ represented in frame k. For instance, the HTM between frames $X_0 Y_0 Z_0$ and $X_B Y_B Z_B$ is given by

$$T_B^0 = \begin{bmatrix} cos(\theta_1) & -sin(\theta_1) & 0 & 0 \\ sin(\theta_1) & cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & H_B \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

Where H_B is the height of the gantry. The position p_{foot}^0 of the foot in respect to the base frame expressed in world-fixed coordinates is

$$\begin{bmatrix} p_{foot}^0 \\ 1 \end{bmatrix} = T_F^0 \begin{bmatrix} p_{foot}^F \\ 1 \end{bmatrix} = T_B^0 T_H^B T_K^H T_F^K \begin{bmatrix} 0_{3\times 1} \\ 1 \end{bmatrix}. \quad (3)$$

The complete kinematic transformations for HOPPY are described in the provided MATLAB-based simulator.

B. Rigid-body dynamics during flight phase

The robot state vector is given by the collection of joint angles $q = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4]^T$ and their time rate of change $\dot{q}$. When the foot is not contacting the ground and the robot is airborne, the equation of motion (EoM) of the robot is given by the well-known *manipulator equation*

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau, \quad (4)$$

which is derived from energy principles using the system's Lagragian as described in [15]. Where $M(q)$ represents the symmetric and positive definite inertia matrix as a function of the configuration, $C(q,\dot{q})$ is the Coriollis and centrifugal forces matrix, $G(q)$ is the vector of torques due to gravity, and τ is the vector of input torques. The system is underactuated because the first two joints are passive ($\tau_1 = \tau_2 = 0$), and thus, there are four overall DoF but only two inputs [16]. The input vector is given by

$$\tau = B_e u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \tau_s(\theta_4) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_H \\ \tau_K \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \tau_s(\theta_4) \end{bmatrix}, \quad (5)$$

where B_e is a selection matrix which maps the input torques u to the full system dynamics and $\tau_s(\theta_4)$ represents the torque exerted by the knee spring as a function of the knee angle θ_4. The robot controller can only select torques for the hip τ_H and knee τ_K in order to hop around the gantry. The spring attached in parallel with the knee alleviates the joint's torque requirements for hopping. The expression for $\tau_s(\theta_4)$ and the rigid-body parameters for each link mass, center of mass (CoM) position, and inertia tensor are provided in the instructions.

C. Actuator dynamics model

Equation (5) assumes that the actuators are perfect torque sources which can generate arbitrary torque profiles. This assumption may be reasonable for simulation, but in reality the actuator dynamics play a major role in the overall system behavior and stability [10]. Sophisticated drivers can rapidly regulate the current i in the motor coils, which is approximately linearly proportional to the output torque $\tau = k_T i$ if the steel saturation is neglected (k_T is the motor constant [7]). However, such drivers are expensive (over \$100) and here we employ the \$25 *Pololu VNH5019* in our kit, which can only control the voltage V across the motor terminals. In this situation, the back-electromotive force (back-EMF) significantly effects the actuator behavior. On the other hand, this limitation provides a valuable opportunity for students to explore how the actuator performance affects the overall dynamics. The model for a brushed and brushless electric motor assuming negligible coil inductance is given by [10]

$$V = R_w i + k_v N \dot{\theta}, \tag{6}$$

$$\ddot{\theta} N^2 I_r = k_T i. \tag{7}$$

Where R_w is the coil resistance, k_v is the speed constant in $\frac{Vs}{rad}$, $\dot{\theta}$ is the joint velocity (after the gear box), I_r is the motor rotor inertia, and N is the gearbox speed reduction ratio. The hip and knee actuators in HOPPY utlize identical electric motors, but with different gear ratios ($N_H = 26.9$ and $N_K = 28.8$). Equation (4) is augmented to include the actuator dynamics using

$$(M(q) + M_r)\ddot{q} + (C(q, \dot{q}) + B_{EMF})\dot{q} + G(q) = \tau. \tag{8}$$

Where M_r is the inertial effect due to rotor inertia and B_{EMF} is the (damping) term due to the back-EMF:

$$M_r = I_r \begin{bmatrix} 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & \begin{bmatrix} N_H^2 & 0 \\ 0 & N_K^2 \end{bmatrix} \end{bmatrix}, \tag{9}$$

$$B_{EMF} = \frac{k_v k_T}{R_w} \begin{bmatrix} 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & \begin{bmatrix} N_H^2 & 0 \\ 0 & N_K^2 \end{bmatrix} \end{bmatrix}, \tag{10}$$

and the updated selection matrix and control inputs are given by

$$B_e = \frac{k_T}{R_w} \begin{bmatrix} 0_{2\times2} \\ \begin{bmatrix} N_H & 0 \\ 0 & N_K \end{bmatrix} \end{bmatrix}, \tag{11}$$

$$u = \begin{bmatrix} V_H & V_K \end{bmatrix}^T, \tag{12}$$

which illustrates that the input to the system is the voltage applied to the motors, not the torque they produce. Actuators with large gearing ratios have high output inertia (due to the N^2 term in (7)) and are usually non-backdrivable because of friction amplification. Thus, we select minimal reduction ratio in HOPPY to mitigate the effect of shock loads in the system during touchdown [8], [18]. Joint friction can have major (nonlinear) effects on the robot dynamics and models for stiction, dry coulomb friction, viscous friction, and more are suggested in [10] and can be included in the model.
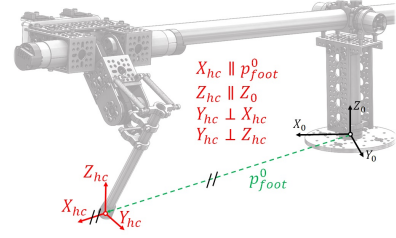

Fig. 3. The frame used for the holonomic constraint for non-slip condition. Frame $X_{hc}Y_{hc}Z_{hc}$ changes every time the robot establishes a new foothold and does not necessarily coincides with foot-fixed frame $X_F Y_F Z_F$.

D. Rigid-body dynamics during stance phase

Equations (4) and (8) are valid when the robot foot is not touching the ground. When contact occurs, the EoM must describe how the ground reaction forces (GRF) affect the robot motion. Contact mechanics are challenging to simulate in complex systems and commercial physics simulation engines, such as *MuJoCo* [17], employ different numerical techniques to approximate them. Two common approximations are know as *soft* and *hard* contact models [19]. The soft contact model approximates the contact mechanics by modelling the ground as a spring and damper. Thus, the foot *must* penetrate the ground to generate contact forces. Because the ground is assumed to have high stiffness values, small deformations render large contact forces, and the solver may encounter numerical instability [2]. In contrast, the hard contact model, which is used in our simulator, does not allow the foot the to penetrate the ground. It includes a *holonomic* constraint which assumes that the foot does not slip in the Y_{hc} and Z_{hc} directions, shown in fig. 3, when in contact with the ground. However, the foot is allowed to slip in the X_{hc} direction, otherwise the system would be overconstrained. The Z_{hc} axis is always vertical, axis X_{hc} is colinear with the line that connects the gantry base and the foot, and Y_{hc} is perpendicular to these. Frame $X_{hc}Y_{hc}Z_{hc}$ changes every time the robot establishes a new foothold and does not necessarily coincides with foot-fixed frame $X_F Y_F Z_F$. We utilize the Jacobian $J_{hc} \in \mathbb{R}^{2\times4}$ to write a holonomic constraint for the non-slip condition

$$J_{hc}\dot{q} = \begin{bmatrix} v_{Yhc} \\ v_{Zhc} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{13}$$

Where v_{Yhc} and v_{Zhc} are the foot linear velocity in the Y_{hc} and Z_{hc} directions. Equation (13) is satisfied if [19]

$$J_{hc}\ddot{q} + \dot{J}_{hc}\dot{q} = 0_{2\times1} \tag{14}$$

is also satisfied. Moreover, when the robot pushes against the ground, it generates reaction forces $F_{GRF} = \begin{bmatrix} F_{Yhc} & F_{Zhc} \end{bmatrix}^T$ which alter the EoM by $J_{hc}^T F_{GRF}$. Finally, the complete stance dynamics is represented by the matrix equation which includes the manipulator equation plus the holonomic constraint:

$$\begin{bmatrix} M(q) & -J_{hc}^T \\ J_{hc} & 0_{2\times2} \end{bmatrix} \begin{bmatrix} \ddot{q} \\ F_{GRF} \end{bmatrix} = \begin{bmatrix} \tau - C(q,\dot{q})\dot{q} - G(q) \\ -\dot{J}_{hc}\dot{q} \end{bmatrix}. \tag{15}$$

Hence, when solving the forward dynamics, the numeric integrator (MATLAB's *ode45()*) takes the input vector u from

the control law and calculates the joints acceleration $\ddot{q}$ and the resultant contact forces F_{GRF} at every iteration.

E. Foot impact model at touchdown

The impact dynamics are challenging to model realistically and they may cause numerical issues for the solver. Our simulator assumes a completely inelastic collision, which means that the foot comes to a complete stop after hitting the ground. And thus, the robot looses energy every time the foot impacts the ground [18]. This assumption implies that the impulsive impact forces remove energy from the system, and the velocity instantaneously changes from $\dot{q}^-$ at time t^- before the impact to $\dot{q}^+$ immediately after the impact at time t^+. However, because this discontinuous changes occurs instantaneously, the joint angles remain unchanged ($q^- = q^+$). To compute the robot velocity after the impact we integrate both sides of equation (15) between t^- and t^+. Because the joint angles and control inputs do not change during impact we obtain [1], [19]

$$M(\dot{q}^+ - \dot{q}^-) = J_{hc}^T F_{imp}, \tag{16}$$

in addition to the constraint due to the inelastic collision $J_{hc}\dot{q}^+ = 0$. We utilize these equations to write the *impact map* which defines the transition between the aerial phase described by the dynamic EOM (4) to the stance phase described by equation (15) [19]:

$$\begin{bmatrix} \dot{q}^+ \\ F_{imp} \end{bmatrix} = \begin{bmatrix} M(q) & -J_{hc}^T \\ J_{hc} & 0_{2\times 2} \end{bmatrix}^{-1} \begin{bmatrix} M(q)\dot{q}^- \\ 0_{2\times 1} \end{bmatrix}. \tag{17}$$

Notice that the impact force F_{imp} is affected by the inverse of the inertia matrix, which is amplified by the reflected inertia of the rotors. Hence, actuators with high gear ratios will create large impact forces, and thus, HOPPY employs minimal gearing [18].

F. Numerical simulation

A *hybrid system* is a dynamical system that exhibits both continuous and discrete behavior. HOPPY displays continuous dynamics during flight phase or stance phase, but experiences discontinuous transitions between the two. To simulate hybrid systems, solvers like MATLAB's *ode45()* utilize guard functions to flag the transitions between continuum phases. For instance, the numerical simulation starts with the aerial phase dynamics and is interrupted when the *touchdown* guard function (foot height) cross a certain threshold (ground height), or: $p_{Zfoot}^0(t) = 0$. Similarly, the robot takes-off the ground when the vertical component of the ground reaction force reaches a value equal to zero: $F_{ZGRF}(t) = 0$. This sequence is shown in fig. 4. The simulator starts in the aerial phase and runs the integration for n hops, which is defined by the number of the transitions between the stance to aerial phase. The simulator is described in detail in the provided instructions.
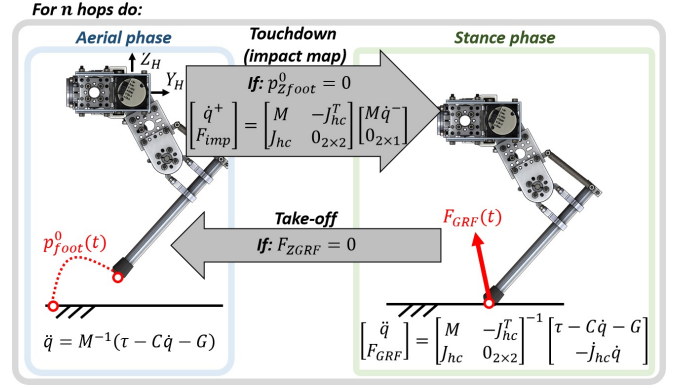

Fig. 4. Simulation loop of the hybrid dynamics of HOPPY for n hops. Continuous dynamics in flight (equation (4)) and stance (equation (15)) are interrupted by discrete events: touchdown and lift-off.

G. Hopping control framework

A challenging aspect of controlling hopping is that the foot experiences little resistance during aerial phase, but *infinite* resistance when touching the ground during stance phase. Hence, the controller must switch between different policies for the two scenarios. During aerial phase, the foot may follow a predefined trajectory to reach a desired stepping location at impact, as shown in fig. 4 left. However, during stance the foot pushes against the ground to apply a desired impulse to the CoM. Thus it is a common practice to perform position control of the leg during swing and force control of the ground contact force during stance. We employ a simple strategy in which, during flight, the leg holds a desired configuration and waits for touchdown, while, during stance, the foot applies a predefined force profile against the ground. An insightful and accessible discussion on hopping control is provided in [9], [12].

During flight phase, HOPPY simply holds a predefined leg configuration and waits for touchdown. To achieve this, we employ a simple task-space Proportional plus Derivative (PD) control using:

$$\tau_{aerial} = J_c^T \left[K_p(p_{ref}^H - p_{foot}^H) - K_d \dot{p}_{foot}^H \right], \tag{18}$$

where $J_c(q) \in \mathbb{R}^{2\times 2}$ is the foot contact point Jacobian, which maps the joint velocities to the linear velocity of the foot in respect to the hip frame $[\dot{y}_{foot}^H \quad \dot{z}_{foot}^H]^T = J_c[\dot{\theta}_H \quad \dot{\theta}_K]^T$. In addition, $K_{p,d}$ are diagonal and positive semi-definite proportional and derivative gain matrices, p_{ref}^H is the desired position of the foot in respect to the hip at touchdown, and p_{foot}^H is its current value. The desired torques τ_{aerial} are mapped using the motor dynamics to the required input voltages u_{aerial}:

$$u_{aerial} = \begin{bmatrix} \frac{R_w}{k_T N_H} & 0 \\ 0 & \frac{R_w}{k_T N_K} \end{bmatrix} \tau_{aerial} + k_v \begin{bmatrix} N_H \hat{\dot{\theta}}_H \\ N_K \hat{\dot{\theta}}_K \end{bmatrix}. \tag{19}$$

However, notice the positive feedback sign before k_v and that $\hat{\dot{\theta}}_{H,K}$ are the *estimated* joint velocities from an observer or filtered derivative. Thus, overestimation of the speed constant k_v or delay in the estimation of the joint velocities may cause oscillations or even instability. For the presented experiments
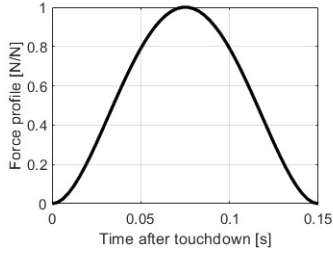
Fig. 5. Prescribed force profile for the horizontal (Y_H) and vertical (Z_H) components of F_{GRF} during a projected stance of $0.15s$. The peak horizontal and vertical forces are modulated to control locomotion speed and hopping height, respectively.

Fig. 6. Achievable torque (τ) and speed ($\dot{\theta}$) of the hip and knee joints. Functions $\phi_{1,2}(\tau, \dot{\theta})$ are defined by the gearbox ratio N, the torque constant k_T, and the maximum current the driver can supply I_{max}. Functions $\phi_{3,4}(\tau, \dot{\theta})$ are defined by the voltage supplied V_{max}, which limits the stall torque $\tau_{stall} = \frac{V_{max} N}{R_w} k_T$ and no-load speed $\dot{\theta}_{NL} \approx \frac{V_{max}}{N} k_v$ achievable by the joint. The operating region is larger for negative work (τ and $\dot{\theta}$ with opposite signals) due to the effect of the back-EMF.

we use a conservative approximation for the speed constant assuming $k_v \approx 0$.

During stance phase, the robot's foot applies a predefined force profile against the ground, which results in an desired net impulse on the robot. The force profile can be generated using simple functions, such as polynomials, in order to be simple to compute online. Here we utilize Bézier polynomials to create a force profile for a projected $0.15s$ stance duration as shown in fig. 5 [1]. The force peak is modulated to control jumping height and forward velocity. To apply the desired force against the ground, we map F_{dGRF} to the desired joint torques using the contact Jacobian similarly to equation (18):

$$\tau_{stance} = J_c^T F_{dGRF}. \tag{20}$$

Equation (20) assumes that the motion of the lightweight leg does not significantly affect the body dynamics [12]. Otherwise, the control law would need to include the torque required to accelerate the leg mechanism using, for instance, Computed Torque Control [15]. Torque is mapped to input voltage using equation (19).

Finally, rapidly switching between the controllers for the stance and aerial phases can cause contact instability issues on the real robot. To smoothly transition between the two we command the input

$$u = \alpha u_{stance} + (1 - \alpha) u_{aerial}, \tag{21}$$

in which α smoothly changes from 0 to 1 after touchdown within a predefined time, typically in the order of $10ms$.

This proposed control strategy is not affected by the leg singular configuration (straight knee) because it employs the transpose of the contact Jacobian, not its inverse. In addition, the simple strategy also does not require solving the robot inverse kinematics.

H. Mechatronics

The control of HOPPY involves fundamental concepts related to Mechatronics education. Including:

- **Embedded systems:** The robot controller runs in an onboard microcontroller (μC) which interfaces with the actuators, sensors, and the host computer.
- **Discrete control:** The μC performs computations and interfaces with peripherals at discrete time iterations. And the deterministic execution of these events is fundamental
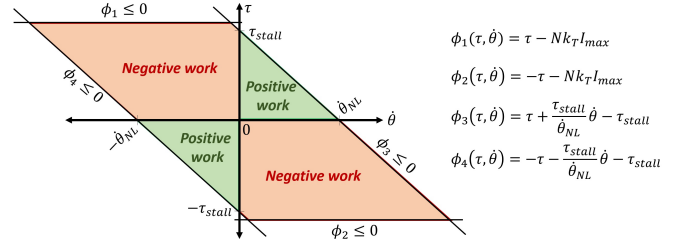
for the implementation of discrete control policies. To effectively control dynamic motions, we target a control rate in the order of $1kHz$ (control loop of $1ms$).

- **Communication protocols:** The μC regulates the motor voltage via Pulse Width Modulation (PWM), receives an analog signal which is proportional to the motor current, a binary signal for the foot contact switch sensor, and employs dedicated counters for the incremental encoders.
- **Signal processing:** The encoder has a coarse resolution of 28 counts per revolution (CPR) and the gearbox has noticeable backlash. To avoid noise amplification we estimate the motor velocity using a filtered derivative $\frac{\dot{\theta}(s)}{\theta(s)} = \frac{\lambda s}{\lambda + s}$ converted to discrete time with a period of $T = 1ms$, where s is the Laplace variable for frequency, and λ is a tunable constant (usually $\lambda \approx 10$). Fast sampling rates are essential to avoid delayed velocity estimation. The analog signal from the motor current estimation is also noisy and requires filtering if used for controls.
- **Saturation:** The achievable joint torques and speeds are limited by the available voltage supply ($V_{max} = 12V$) and the maximum current that the driver can provide ($I_{max} = 30A$) [10]. The operating region of the motors is depicted in fig. 6.

III. KIT DESIGN AND COMPONENTS

HOPPY is designed exclusively with off-the-shelf components to lower its cost and enable modularity, customization, and facilitate maintenance. In addition, it is lightweight (total weight about $3.8kg$), portable, and mechanically robust. All load-bearing components are made of metal parts from https://gobilda.com/. We intentionally avoid the use of plastic 3D printed parts due to their short working life under impact loads. Because of the kit's durability, only the initial cost for setting up the course is required and the equipment can be reutilized in future semesters with minimal maintenance costs. Both motors are placed near the hip joint to reduce the moving inertia of the leg and enable fast swing motions. The second actuator is mounted coaxially with the hip joint and drives the knee joint through a timing belt. The motors have minimal gearing ratio in order to be backdrivable. To reduce the torque
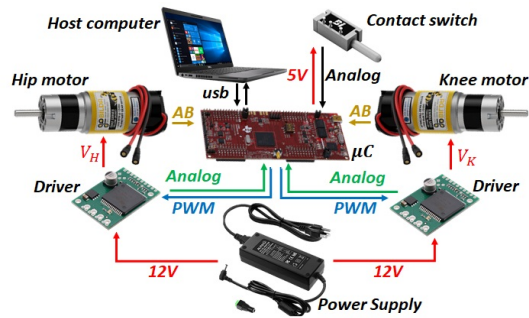
Fig. 7. Electrical components of the kit and communication diagram.

requirements for hopping, springs are added in parallel with the knee and a counterweight is attached at the opposite end of the gantry as shown in fig. 1. We employ cheap $2.3kg$ ($5lb$) weights typically utilized for gymnastics. The addition of an excessively heavy counterweight reduces the achievable frictional forces between the foot and the ground, and the robot is more likely to slip.

The diagram of electrical connections and communications protocols is depicted in fig. 7. The user computer communicates with the embedded μC (*Texas Instruments LAUNCHXL-F28379D*) via a usb port. The μC commands the desired voltage to the motors (*goBilda 5202-0002-0027* for the hip and *0019* for the knee) via PWM to the drivers (*Pololu VNH5019*), and receives analog signals proportional to the motor current and the pulse counts from the incremental encoders. The motor drivers are powered by a 12V power supply (*ALITOVE 12Volt 10 Amp 120W*). A linear potentiometer with spring return (*TT Electronics 404R1KL1.0)* attached in parallel with the foot works as a foot contact switch.

The kit available at https://github.com/RoboDesignLab/HOPPY-Project is composed of:
1) the solid model in *SolidWorks* and .STEP formats;
2) the video instructions for mechanical assembling;
3) the electrical wiring diagram
4) the dynamic simulator code and instructions;
5) the complete list of components and quantities; and
6) the basic code for the microcontroller.

IV. MODIFICATIONS, LIMITATIONS, AND SAFETY

The kit is designed to facilitate the adjustment of physical parameters and replacement of components. For instance, gearboxes with ratios of $3.7:1$ to $188:1$ can be used in either the hip or knee. The length of the lower leg is continually tunable between $60mm$ and $260mm$. Additional springs can be added (or removed) to the knee joint if necessary. The gantry length is continually adjustable between $0.1m$ and $1.1m$ and its height has multiple discrete options between $0.2m$ and $1.1m$. The dynamic effect of the counterweight can be adjusted by shifting its position further away from the gantry joint or adding more mass.

However, the kit also presents performance limitations. For instance, voltage control of the motor is not ideal as it hinders the precise regulation of ground reaction forces at high speed. The coarse resolution of the encoders and the backlash in

the gearbox also degrades position tracking performance. The hobby-grade brushed motors have limited torque density (peak torque capability divided by unit mass), and thus the robot *will not* be able to hop without the aid of the springs and the counterweight. In addition, the simulator assumes simple dynamic models for impact and contact, and neglects friction, structural compliance, vibrations, and foot slip. Due to these limitations, it is *unlikely* that the physical robot will behave exactly like the simulator predicts. However, the simulator is a valuable tool to obtain insights about the behavior of the real robot, it can be used for preliminary tuning of controllers before implementation in the hardware, and is a fundamental tool for teaching.

When experimenting with HOPPY (or any dynamic robot) some safety guidelines must be followed. Students should wear safety goggles and clear the robot's path. It is advised to *constantly* check the temperature of the motor's armature. If they are too hot to be touched, the experiments should be interrupted until cool down. Aggressive operations (high torque and speeds) within negative work regimes should be done carefully because electric motors regenerate part of the input energy back into the power supply when backdriven [13]. And thus, utilizing a battery is more appropriated if considerable negative work is expected. For instance, a conventional $12V$ car battery can be used both as power supply and a mechanical counterweight. Finally, it is advised to often check for loose components due to the constant impacts and vibrations.

If the budget allows, some components of HOPPY can be improved. For instance, (i) employing encoders with finer resolution around 12bits (4096 counts per revolution) would be ideal. This would improve position control and joint velocity estimation. (ii) Implementation of motor drivers which can perform high-speed current control, such as the *Advanced Motion Control AZBDC12A8*. This feature would enable precise torque control and considerably improve force tracking during stance. An even better solution would be (iii) the utilization of brushless motors similar to those employed in [3], [7]. These have much superior torque density and would enable hopping without the aid of springs or counterweights. (iv) Additional sensors (encoders) could be added to the gantry joints ($\theta_{1,2}$) or an Inertial Measurement Unit (IMU) can be mounted on the robot for state estimation [4]. (v) Data logging can be achieved using external loggers such as the *SparkFun OpenLog*. And finally, (vi) to enable unrestricted rotations around the gantry, a battery can be used to power the robot and a USB slip ring can be included for communication between the μC and the host's computer.

V. RESULTS AND DISCUSSION

A snapshot of the output of the MATLAB-based simulator is shown in fig. 8. It displays an animation of the robot during hopping (including a view of the robot's Sagittal plane) with a trace of the hip joint and foot spatial trajectories. The visualizer also shows the hopping velocity, the joint states, required torques, and the working regions of the actuators to check for saturation. Areas shaded in red represent stance phases. More information about the simulator is provided in the accompanying instructions.
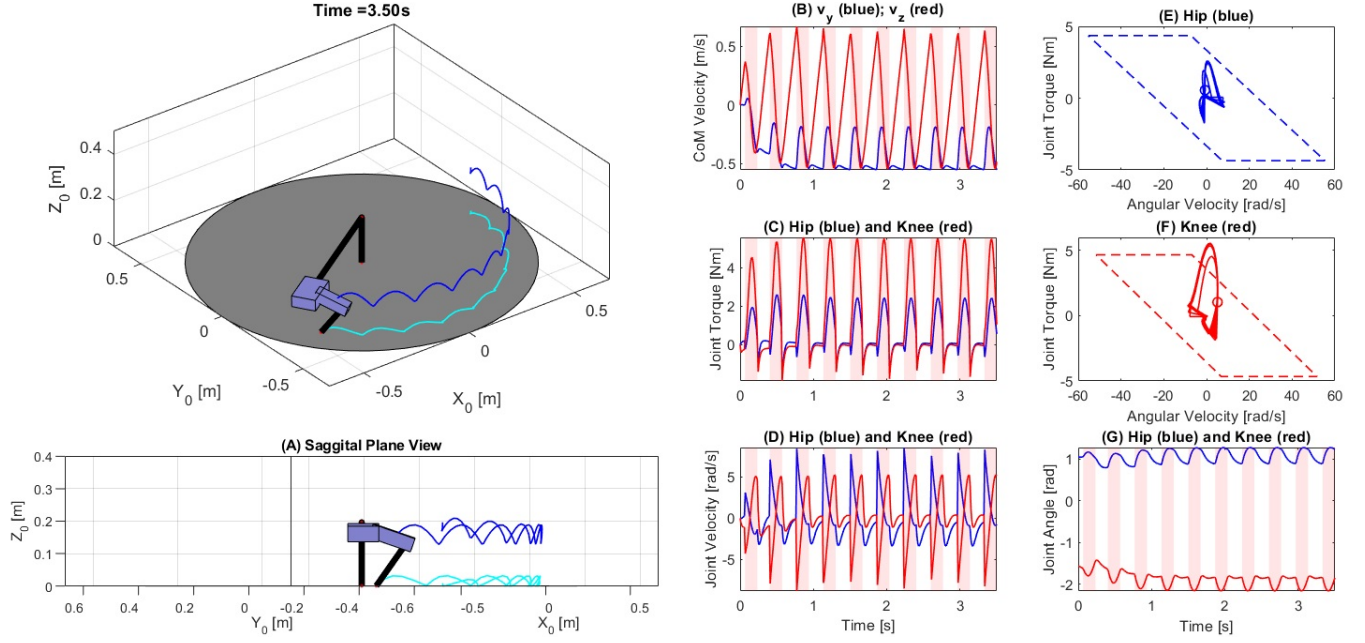
Fig. 8. Screenshot of the output screen of the simulator. It shows the robot animation in isometric and Sagittal plane views along with the trajectories of the hip joint in blue and the foot in cyan. Plot (B) shows the horizontal and vertical tangential velocities of the hip base around the gantry. Areas shaded in red represent stance phases. Hip and knee joint torques are shown in (D) and their angular velocities in (D). Plots (E) and (F) display the limiting polygon for the hip and knee actuators along with the torque versus velocity trajectory. Feasible trajectories must be always contained inside the polygon. Finally, plot (G) shows the hip an knee joint angles over time.

For the experiments in companion video (https://youtu.be/_lbKIpiRWKI), we employed the controller described in section II-G using aerial phase gains of $K_p = 500[1_{2\times2}]$ and $K_s = 50[1_{2\times2}]$, the stance duration $T_s = 0.15s$, and the peak vertical force commanded of $80N$. Where $1_{2\times2}$ is a 2×2 identity matrix. The peak horizontal force was modulated to control the locomotion speed. Fig. 9-(A) to (D) shows the data of robot accelerating from about $0.65m/s$ to around $1.2m/s$ by increasing the horizontal force peak. The joint torques are estimated using the measured motor current ($\tau = ik_T N$). The saturation occurs when the requested input voltage is above the supplied $12V$. Fig. 9-(D) shows the stance duration, measured from the contact switch, which decreases according to the locomotion speed. Notice that at fast speeds, the robot touches the ground for less that $100ms$. Finally, 9-(E) shows the achieved steady-state hopping velocities according to the commanded peak horizontal force. We note that the locomotion speed is likely overestimated because it was computed using the relative velocity between the foot and the hip during stance, assuming the the foot does not slip during contact.

HOPPY was used in a semester-long project in the *ME446 Robot Dynamics and Control* course at UIUC. The course is attended by senior undergraduate and first-year graduate students from the Departments of Mechanical Science and Engineering, Electrical and Computer Engineering, and Industrial & Enterprise Systems Engineering. Ten cross-departmental teams of four to five students each received their own leg (hip base of link 2 plus links 3 and 4), and all teams shared six gantries (fixed base plus links 1 and 2). During the first half of the course, students derived the kinematic and dynamic model of the robot, developed different controllers, and created a simple version of the robot simulator in MATLAB. They assembled their kits half-way through the semester and performed experiments with the robot on the second half. After developing their own dynamic simulator, students were provided with our code for comparison. The results from experiments with the real robot from one of the teams are shown in the snapshots of fig. 10. The team developed a custom foot contact sensor to detect stance phase. The robot was able to regulate hopping speed, overcome small obstacles, and mitigate disturbances from gentle kicks.

VI. CONCLUSION AND FUTURE WORK

This letter introduced HOPPY, an open-source, low-cost, robust, and modular kit for robotics education. The kit lowers the barrier for studying dynamic robot behaviors and legged locomotion with real systems. The control of dynamic motions present unique challenges to the robot software and hardware, and these are often overlooked in conventional robotics courses, even those with hands-on activities. Here we describe the topics which can be explored using the kit, list its components, discuss best practices for implementation, and suggest further improvements. HOPPY was utilized as the topic of a semester-long project for an first year graduate-level course at UIUC. Students provided an overwhelmingly positive feedback from the hands-on activities during the course. The instructors will continue to improve the kit and course content for upcoming semesters. To nurture active learning, future activities will include a friendly competition between teams to elect the fastest robot and the most energy-efficient robot. Speed is estimated by timing complete laps
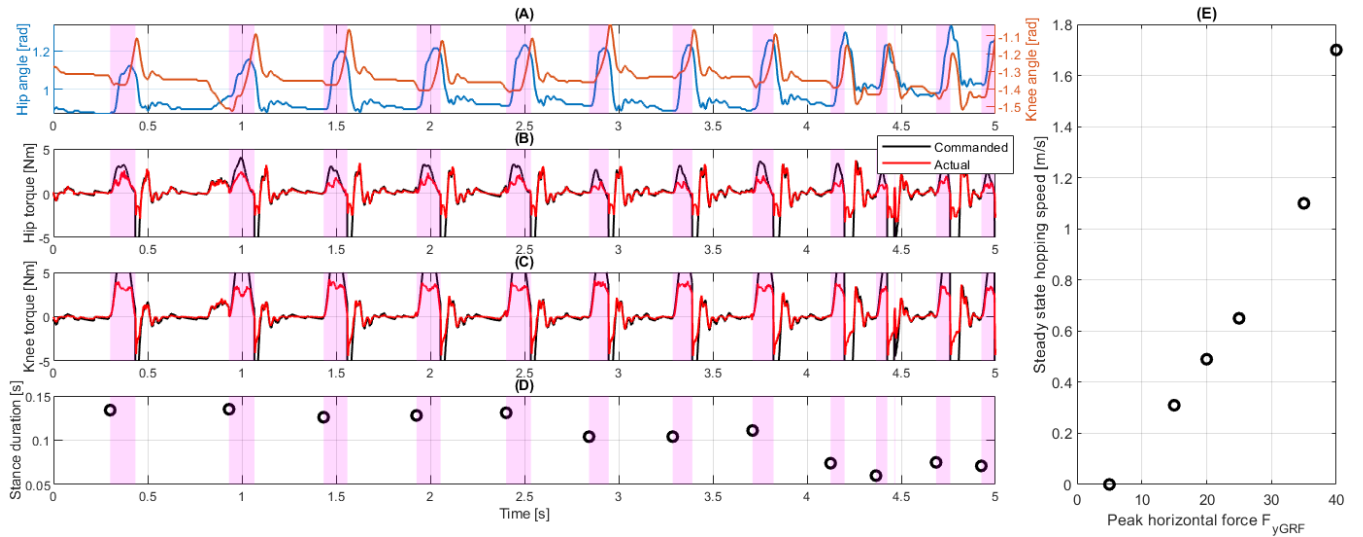
Fig. 9. Experimental data with HOPPY speeding up from around $0.65m/s$ to $1.2m/s$. Shaded magenta areas represent stance periods. (A) Hip (θ_3) joint angles in the left vertical axis and knee (θ_4) angle in right axis. (B) Hip joint commanded torque (black) and actual torque (red) estimated from the measured current. (C) Knee joint torque. (D) Stance phase duration (duration of following shaded areas). And (E) steady-state hopping speed as a function of the commanded peak horizontal contact force F_{yGFP}.

Fig. 10. Snapshots of the hopping experiment with HOPPY by one of the student teams (https://youtu.be/6O6czr0YyC8).

around the gantry. Energy efficiency is determined by the Cost of Transport, which is proportional to the ratio between the total electrical power consumed during one hopping cycle and the average translation speed [13]. Further improvements of the kit will be made open-source.

REFERENCES

[1] Y. Ding and H. Park. Design and experimental implementation of a quasi-direct-drive leg for optimized jumping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 300–305, Sep. 2017.

[2] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2007.

[3] 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*, 5(2):3650–3657, 2020.

[4] Ross Hartley, Maani Ghaffari Jadidi, Jessy W. Grizzle, and Ryan M. Eustice. Contact-aided invariant extended kalman filtering for legged robot state estimation. *CoRR*, abs/1805.10410, 2018.

[5] IEEE Spectrum. Kiva systems: Three engineers, hundreds of robots, one warehouse, 2008.

[6] IEEE Spectrum. Darpa robotics challenge: A compilation of robots falling down, 2015.

[7] 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)*, 2019.

[8] G. Kenneally, A. De, and D. E. Koditschek. Design principles for a family of direct-drive legged robots. *IEEE Robotics and Automation Letters*, 1(2):900–907, July 2016.

[9] Jeff Koechling and H. Marc. How fast can a legged robot run? In *Robots and Biological Systems: Towards a New Bionics?* Springer Berlin Heidelberg, 1993.

[10] Kevin Lynch and Frank Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, USA, 1st edition, 2017.

[11] M. Orta Martinez, C. M. Nunez, T. Liao, T. K. Morimoto, and A. M. Okamura. Evolution and analysis of hapkit: An open-source haptic device for educational applications. *IEEE Transactions on Haptics*, 13(2):354–367, 2020.

[12] Marc H. Raibert. *Legged Robots That Balance*. Massachusetts Institute of Technology, USA, 1986.

[13] S. Seok, A. Wang, Meng Yee Chuah, D. Otten, J. Lang, and S. Kim. Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot. In *2013 IEEE International Conference on Robotics and Automation*, pages 3307–3312, May 2013.

[14] S. D. Sheppard, K. Macatangay, A. Colby, and W. M. Sullivan. *Educating Engineers: Designing for the Future of the Field*. Carnegie Foundation for the Advancement of Teaching, Jossey-Bass, USA, 2009.

[15] Mark W. Spong. *Robot Dynamics and Control*. John Wiley and Sons, Inc., USA, 1st edition, 1989.

[16] Russ Tedrake. Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation (course notes for mit 6.832). downloaded on 09/09/2020 from http://underactuated.mit.edu/.

[17] E. Todorov. Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in mujoco. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6054–6061, 2014.

[18] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim. Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots. *IEEE Transactions on Robotics*, 33(3):509–522, June 2017.

[19] ER Westervelt, JW Grizzle, C Chevallereau, JH Choi, and B Morris. *Feedback control of dynamic bipedal robot locomotion*. CRC press., 2007.