HOPPY: The Technical Manual

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

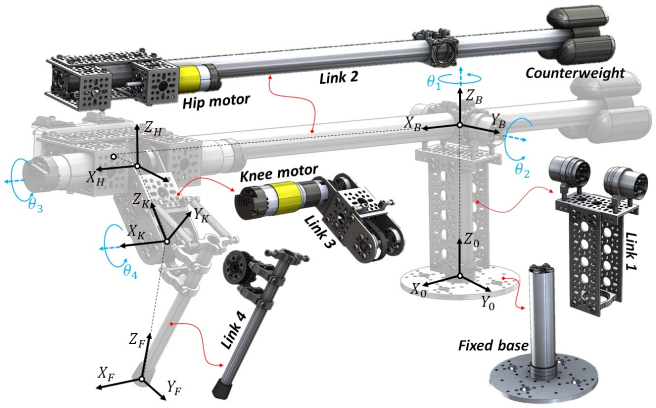

Fig. 1. Kinematic transformations and the five rigid-bodies which compose HOPPY. The first two joints (θ_1 and θ_2) are passive while joints three and four are driven by electric motors.

I. 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* [12]. 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. 1. 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 $X_0Y_0Z_0$ 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 I-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 [12]. The transformation between coordinate frames is described by the Homogeneous Transformation Matrices (HTM) given by T^k_{k+1}, which converts a vector p^{k+1} expressed in coordinate frame $k+1$ into a vector p^k expressed in coordinate frame k [8]:

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

Where the term $0_{1\times 3}$ represents a matrix of zeros with 1 rows and 3 columns. The HTM is composed of a rotation matrix $R^k_{k+1} \in SO(3)$ and a Cartesian translation $d^k \in \mathbb{R}^3$ represented in frame k. The position p^0_{foot} of the foot in respect to the base frame expressed in world-fixed coordinates is

$$\begin{bmatrix} p^0_{foot} \\ 1 \end{bmatrix} = T^0_F \begin{bmatrix} p^F_{foot} \\ 1 \end{bmatrix} = T^0_B T^B_H T^H_K T^K_F \begin{bmatrix} 0_{3\times 1} \\ 1 \end{bmatrix}. \quad (2)$$

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*

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau, \quad (3)$$

which is derived from energy principles using the system's Lagragian as described in [8]. 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 [9]. The input vector is given by

$$\tau = B_e u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \tau_s(\theta_4) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_H \\ \tau_K \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \tau_s(\theta_4) \end{bmatrix}, \quad (4)$$

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

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

mass (CoM) position, and inertia tensor are provided in the instructions.

C. Actuator dynamics model

Equation (4) 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 [6]. 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 [3]). 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 [6]

$$V = R_w i + k_v N \dot{\theta}, \tag{5}$$

$$\ddot{\theta} N^2 I_r = k_T i. \tag{6}$$

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 (3) is augmented to include the actuator dynamics using

$$(M(q) + M_r)\ddot{q} + (C(q,\dot{q}) + B_{EMF})\dot{q} + G(q) = \tau. \tag{7}$$

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

$$M_r = I_r \begin{bmatrix} 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & \begin{bmatrix} N_H^2 & 0 \\ 0 & N_K^2 \end{bmatrix} \end{bmatrix}, \tag{8}$$

$$B_{EMF} = \frac{k_v k_T}{R_w} \begin{bmatrix} 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & \begin{bmatrix} N_H^2 & 0 \\ 0 & N_K^2 \end{bmatrix} \end{bmatrix}, \tag{9}$$

and the updated selection matrix and control inputs are given by

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

$$u = \begin{bmatrix} V_H & V_K \end{bmatrix}^T, \tag{11}$$

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 (6)) 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 [4], [11]. 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 [6] and can be included in the model.

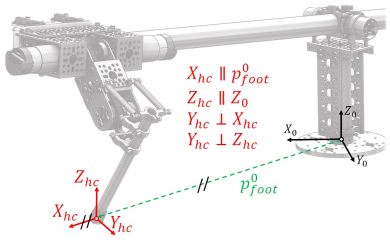

Fig. 2. The frame used for the holonomic constraint for non-slip condition. Frame $X_{hc}Y_{hc}Z_{hc}$ changes every time the robot establishes a new foothold and does not necessarily coincides with foot-fixed frame $X_F Y_F Z_F$.

D. Rigid-body dynamics during stance phase

Equations (3) and (7) 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* [10], employ different numerical techniques to approximate them. Two common approximations are know as *soft* and *hard* contact models [12]. 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. 2, 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

$$J_{hc}\dot{q} = \begin{bmatrix} v_{Yhc} \\ v_{Zhc} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{12}$$

Where v_{Yhc} and v_{Zhc} are the foot linear velocity in the Y_{hc} and Z_{hc} directions. Equation (12) is satisfied if [12]

$$J_{hc}\ddot{q} + \dot{J}_{hc}\dot{q} = 0_{2\times1} \tag{13}$$

is also satisfied. Moreover, when the robot pushes against the ground, it generates reaction forces $F_{GRF} = [F_{Yhc} \quad F_{Zhc}]^T$ 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:

$$\begin{bmatrix} M(q) & -J_{hc}^T \\ J_{hc} & 0_{2\times 2} \end{bmatrix} \begin{bmatrix} \ddot{q} \\ F_{GRF} \end{bmatrix} = \begin{bmatrix} \tau - C(q,\dot{q})\dot{q} - G(q) \\ -\dot{J}_{hc}\dot{q} \end{bmatrix}. \quad (14)$$

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 [11]. 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 (14) between t^- and t^+. Because the joint angles and control inputs do not change during impact we obtain [1], [12]

$$M(\dot{q}^+ - \dot{q}^-) = J_{hc}^T F_{imp}, \quad (15)$$

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 (3) to the stance phase described by equation (14) [12]:

$$\begin{bmatrix} \dot{q}^+ \\ F_{imp} \end{bmatrix} = \begin{bmatrix} M(q) & -J_{hc}^T \\ J_{hc} & 0_{2\times 2} \end{bmatrix}^{-1} \begin{bmatrix} M(q)\dot{q}^- \\ 0_{2\times 1} \end{bmatrix}. \quad (16)$$

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 [11].

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: $p^0_{Zfoot}(t) = 0$. 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. 3. The simulator starts in the aerial phase and runs the integration

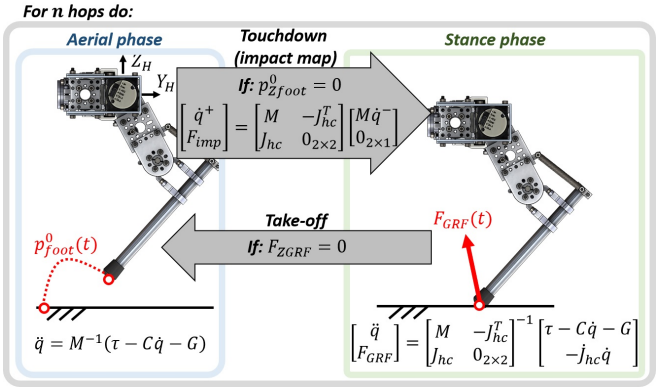

Fig. 3. Simulation loop of the hybrid dynamics of HOPPY for n hops. Continuous dynamics in flight (equation (3)) and stance (equation (14)) are interrupted by discrete events: touchdown and lift-off.

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.

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. 3 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 [5], [7].

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:

$$\tau_{aerial} = J_c^T \left[K_p(p^H_{ref} - p^H_{foot}) - K_d \dot{p}^H_{foot} \right], \quad (17)$$

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 $[\dot{y}^H_{foot} \quad \dot{z}^H_{foot}]^T = J_c[\dot{\theta}_H \quad \dot{\theta}_K]^T$. In addition, $K_{p,d}$ are diagonal and positive semi-definite proportional and derivative gain matrices, p^H_{ref} is the desired position of the foot in respect to the hip at touchdown, and p^H_{foot} is its current value. The desired torques τ_{aerial} are mapped using the motor dynamics to the required input voltages u_{aerial}:

$$u_{aerial} = \begin{bmatrix} \frac{R_w}{k_T N_H} & 0 \\ 0 & \frac{R_w}{k_T N_K} \end{bmatrix} \tau_{aerial} + k_v \begin{bmatrix} N_H \hat{\dot{\theta}}_H \\ N_K \hat{\dot{\theta}}_K \end{bmatrix}. \quad (18)$$

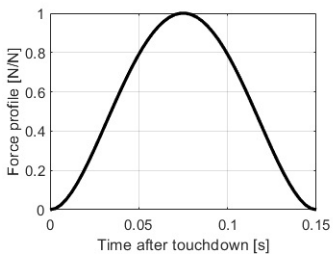

Fig. 4. Prescribed force profile for the horizontal (Y_H) and vertical (Z_H) components of F_{GRF} during a projected stance of $0.15s$. The peak horizontal and vertical forces are modulated to control locomotion speed and hopping height, respectively.

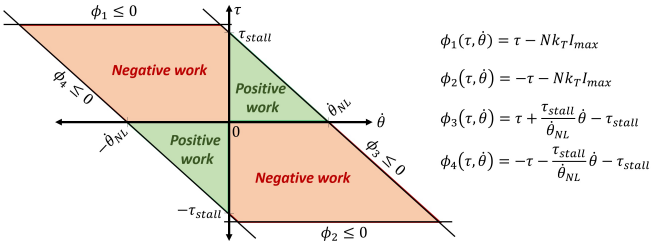

Fig. 5. Achievable torque (τ) and speed ($\dot{\theta}$) of the hip and knee joints. Functions $\phi_{1,2}(\tau,\dot{\theta})$ are defined by the gearbox ratio N, the torque constant k_T, and the maximum current the driver can supply I_{max}. Functions $\phi_{3,4}(\tau,\dot{\theta})$ are defined by the 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.

However, notice the positive feedback sign before k_v and that $\widehat{\dot{\theta}}_{H,K}$ 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 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. 4 [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 (17):

$$\tau_{stance} = J_c^T F_{dGRF}. \tag{19}$$

Equation (19) assumes that the motion of the lightweight leg does not significantly affect the body dynamics [7]. Otherwise, the control law would need to include the torque required to accelerate the leg mechanism using, for instance, Computed Torque Control [8]. Torque is mapped to input voltage using equation (18).

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

$$u = \alpha u_{stance} + (1-\alpha)u_{aerial}, \tag{20}$$

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.
- **Input 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 ($I_{max} = 30A$) [6]. The operating region of the motors is depicted in fig. 5.

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] 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.

[4] 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.

[5] 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.

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

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

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

[9] 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/.

[10] 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.

[11] 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.

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