Autonomous Bikebot Control for Crossing Obstacles With Assistive Leg Impulsive Actuation

Feng Han, *Graduate Student Member, IEEE*, Xinyan Huang, Zenghao Wang, Jingang Yi, *Senior Member, IEEE*, and Tao Liu, *Senior Member, IEEE*

***Abstract*—As a single-track mobile platform, bikebot (i.e., bicycle-based robot) has attractive capability to navigate through narrow, off-road terrain with high speed. However, running crossing step-like obstacles creates challenges for intrinsically unstable, underactuated bikebots. This article presents a novel autonomous bikebot control with two assistive legs to navigate crossing obstacles. The proposed design integrates the external/internal convertible form-based motion control with leg-assisted impulse actuation. The leg–terrain interaction generates impulsive torques to help maintain navigation and balance when running across obstacles. The control performance is analyzed and guaranteed. The experimental results confirm that under the control design, the bikebot can smoothly run crossing multiple step-like obstacles with heights more than one-third of the wheel radius. The comparison results demonstrate the superior performance than those under only the velocity and steering control without leg assistive actuation.**

***Index Terms*—Balance control, bicycle dynamics, impulse control, robot motion control, robot navigation.**

I. Introduction

As A single-track mobile platform, bikebot (i.e., bicycle-based robot) offers attractive navigation capability, such as high speed and high energy efficiency on off-road terrains [1]. Comparing with double-track platform, bikebot has high agility but suffers intrinsic instability and underactuation disadvantages. One major control challenge of underactuated bikebot lies in the nonminimum phase, unstable roll dynamics [2]. Modeling and control of single-track bikebots attracted extensive attention in past decades [3]–[5], and various autonomous bikebots have been developed and demonstrated [6]–[8]. An external and internal convertible (EIC) form was presented for bikebot dynamics [4], and the EIC-based control was successfully demonstrated [5], [9], [10]. Besides steering and velocity control, additional assistive actuation, such as reaction wheels [7], [11], gyroscopic balancer [12], [13], or inverted pendulum [14], was installed on the bikebot to enhance balance and tracking performance.

In all of the abovementioned studies, bikebots were considered to navigate on flat firm terrains. For outdoor off-road applications, bikebots often need to run crossing step-like obstacles, such as curbs and steps. The major challenges for bikebots to navigate across these terrains include complex wheel–obstacle interactions and limited stabilization actuation [14]. It is well known that legged locomotion has advantages to navigate on uneven terrains [15], [16], and using both wheel and leg modalities enables mobile robots to adapt to different operation environments and expand their workspace [17], [18]. Inspired by this observation, we propose a leg-assisted impulse control of bikebot to help balance and run across step-like obstacles.

When running across an obstacle, the wheel–obstacle impact disturbs the bikebot stability, and the steering and velocity control only provides limited capability to maintain balance. In this case, the assistive legs can provide additional balance torques. Due to high moving speed, the leg–terrain contact should be kept as short as possible to minimize its negative impact on bikebot motion. Therefore, a leg–terrain impulsive interaction is desirable for bikebot balance assistance. Impulse control has been used to regain the stability of dynamical systems and enlarge the region of attraction in robotic applications [19]. In [20], an impulse control algorithm was developed for a single degree-of-freedom (DOF) Hamiltonian system, such as the cart–pendulum example. One main difference in this work is the integration of the EIC-based control with the impulse control for underactuated balance robots, such as bikebot.

In this article, we present a novel control design of an autonomous bikebot with two assistive legs for running across obstacles. A dynamic model of the bikebot system is first presented. The wheel–obstacle impact is modeled and its effects

Manuscript received 28 December 2021; revised 28 March 2022; accepted 29 April 2022. Date of publication 25 May 2022; date of current version 16 August 2022. Recommended by Technical Editor Y. Wan and Senior Editor X. Chen. The work of Jingang Yi was supported in part by the U.S. NSF under Grant CNS-1932370. The work of Tao Liu was supported by NSFC under Grant 52175033 and Grant U21A20120. *(Feng Han and Xinyan Huang equally contributed to this work.) (Corresponding authors: Jingang Yi; Tao Liu.)*

Feng Han and Jingang Yi are with the Department of Mechanical and Aerospace Engineering, Rutgers University, Piscataway, NJ 08854 USA (e-mail: fh233@scarletmail.rutgers.edu; jgyi@rutgers.edu).

Xinyan Huang, Zenghao Wang, and Tao Liu are with the State Key Lab of Fluid Power and Mechatronic Systems and the School of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: hxinyan@zju.edu.cn; zenghao_wang@zju.edu.cn; liutao@zju.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMECH.2022.3172909.

Digital Object Identifier 10.1109/TMECH.2022.3172909

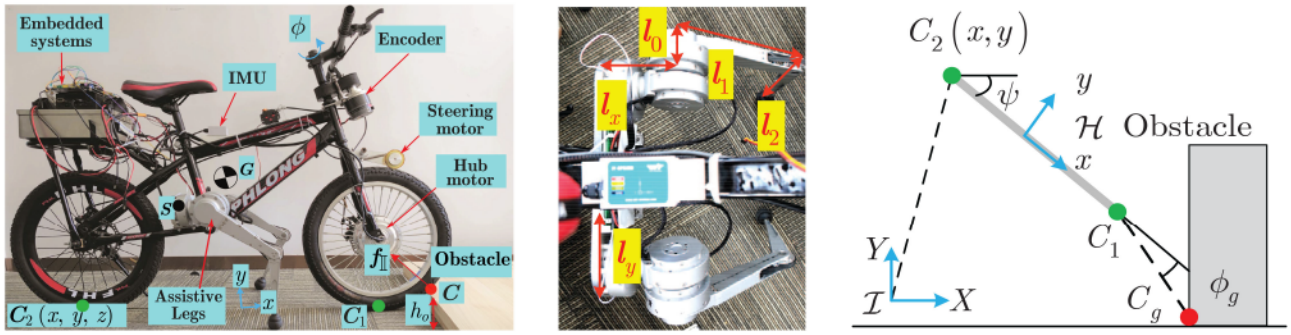

Fig. 1. (a) Prototype of the bikebot system with two assistive legs. (b) Kinematic configuration of the assistive legs. (c) Schematic of the wheel–obstacle interactions.

on bikebot dynamics are estimated. A machine learning-based method is used to enhance the estimated postwheel–obstacle impact motion by using onboard inertial measurement unit (IMU) measurements. The leg–terrain impulsive control is designed through an optimization-based method for real-time implementation. The integration and performance analysis of the impulse control and the EIC-based design are discussed. We demonstrate the control design with extensive experiments. Compared with other assistive devices (e.g., gyroscopic balancers) that often suffer limited actuation, the leg impulsive control provides continuous assistive torques. The main contributions of the work are twofold. First, we consider the use of assistive leg and impulse control for wheeled robot to navigate crossing obstacles. To the best of our knowledge, no reported research work explicitly discusses the mechatronic design for helping unstable mobile robots cross obstacles. The integrated wheel–leg locomotion control is also novel and potentially enables new bikebot applications. Second, the impulse control is designed though an optimization approach, and it has an advantage to integrate with the previously developed trajectory tracking and balance control for underactuated robots.

The rest of this article is organized as follows. In Section II, we present the dynamics and control of the bikebot system. Section III discusses the wheel–obstacle impact and estimation. The impulse control design is discussed in Section IV and experiments are presented in Section V. Finally, Section VI concludes this article.

II. Systems Dynamics and EIC-Based Control

A. Problem Statement

Fig. 1(a) shows a prototype of the autonomous bikebot with two assistive legs. The bikebot is driven by a wheel hub motor, and the steering actuation is powered by a dc motor. Two 3-DOF assistive legs are installed at the bikebot's lower frame with one leg on each side symmetrically. As shown in the figure, when moving across a step-like obstacle (e.g., wooden board), the wheel–obstacle impact generates a large disturbance for the bikebot motion and the platform might lose stability. The problem considered is to design the leg impulsive torque control together with the steering and velocity control to maintain the balance and trajectory tracking when running across obstacles.

B. System Dynamics and EIC-Based Control

Fig. 1 shows the schematic of the systems configuration. The bikebot roll, yaw, and steering angles are denoted as ψ, φ_b, and ϕ, respectively. Three coordinate frames are used in systems modeling: inertial frame $\mathcal{I}$, body frame $\mathcal{B}$, and moving frame $\mathcal{H}$. Frame $\mathcal{I}$ is fixed on the ground with the Z-axis perpendicular to the ground, while the origin of $\mathcal{H}$ is located at the middle point of line C_1C_2, with the z-axis pointing upward and x-axis aligned with C_1C_2, where C_1 and C_2 are the front and rear wheel contact points, respectively. Frame $\mathcal{B}$ is obtained from $\mathcal{H}$ by rotating φ_b around the x-axis with the same origin. The center of mass of the bikebot is denoted as G and its coordinate in $\mathcal{B}$ is $[l_G\ 0\ h_G]^T$.

1) Bikebot Model and EIC-Based Control: The bikebot planar motion is captured by the C_2 position vector $\boldsymbol{r} = [x\ y]^T$ in $\mathcal{I}$. The nonholonomic constraint at C_2 gives $\dot{x} = v\mathrm{c}_\psi$, $\dot{y} = v\mathrm{s}_\psi$, where v is the bikebot velocity magnitude and notations $\mathrm{c}_\psi := \cos\psi$ and $\mathrm{s}_\psi := \sin\psi$ are used for ψ and other angles. Taking the third-order derivative of $\boldsymbol{r}$ produces

$$\boldsymbol{r}^{(3)} = \begin{bmatrix} x^{(3)} \\ y^{(3)} \end{bmatrix} = -\underbrace{\begin{bmatrix} v\dot{\psi}\,\mathrm{c}_\psi + 2\dot{v}\,\mathrm{s}_\psi \\ v\dot{\psi}\,\mathrm{s}_\psi - 2\dot{v}\,\mathrm{c}_\psi \end{bmatrix}}_{\boldsymbol{R}_\psi} \dot{\psi} + \underbrace{\begin{bmatrix} \mathrm{c}_\psi & -v\,\mathrm{s}_\psi \\ \mathrm{s}_\psi & v\,\mathrm{c}_\psi \end{bmatrix}}_{\boldsymbol{K}_\psi} \boldsymbol{u} \tag{1}$$

where $\boldsymbol{u} = [u_v\ u_\psi]^T$, $u_v = \ddot{v}$, and $u_\psi = \ddot{\psi}$. From [21], the yaw angular rate is calculated as $\dot{\psi} = \frac{v\,\mathrm{c}_\varepsilon}{l\,\mathrm{c}_{\varphi_b}} \tan\phi$, and u_ψ is

$$u_\psi = \frac{\dot{v}\tan\phi\mathrm{c}_\varepsilon}{l\mathrm{c}_{\varphi_b}} + \frac{v\,\mathrm{c}_\varepsilon}{l\,\mathrm{c}_{\varphi_b}}(\sec^2_\phi\,\dot{\phi} + \tan\phi\tan\varphi_b\dot{\varphi}_b) \tag{2}$$

where l is the wheelbase and ε is the steering caster angle. The roll motion of the bikebot is captured as [21]

$$J_t\ddot{\varphi}_b = f_1(\varphi_b) + h(\varphi_b)u_\psi \tag{3}$$

where $f_1(\varphi_b) = m_b h_G\dot{\psi}\mathrm{c}_{\varphi_b}v + m_b h_G^2\dot{\psi}^2\mathrm{s}_{\varphi_b}\mathrm{c}_{\varphi_b} + mgh_G\mathrm{s}_{\varphi_b}$, $h(\varphi_b) = m_b h(l_G + l/2)\mathrm{c}_{\varphi_b}$, $J_t = J_b + m_b h_G^2$, and J_b and m_b are the mass moment of inertia around the x-axis and mass of the bikebot, respectively. Substituting (2) into (3) and collecting

terms in $\tan\phi$, we obtain the steering-induced balance torque

$$\tau_s = \frac{m_b h_G \, \mathrm{c}_\varepsilon \, v}{l}\left[\frac{(l_G + l/2)\dot{\varphi}_b}{l}\tan\varphi_b + \frac{l_G \dot{v}}{v} - v\right]\tan\phi.$$

Using τ_s, the roll motion model (3) becomes

$$J_t \ddot{\varphi}_b = f_2(\varphi_b) + \tau_s \tag{4}$$

where $f_2(\varphi_b) = m g h_G \, \mathrm{s}_{\varphi_b} + m_b h_G^2 \frac{v^2 \mathrm{c}_\varepsilon^2}{l^2}\tan\varphi_b \tan^2\phi + m_b h_G (l_G + l/2)\frac{v \mathrm{c}_\varepsilon}{l}\sec^2_\phi \dot{\phi}$.

For a given trajectory $\boldsymbol{r}_d = [x_d \; y_d]^T$, the tracking control

$$\boldsymbol{u} = \boldsymbol{K}_\psi^{-1}(\boldsymbol{R}_\psi + \boldsymbol{u}_r), \boldsymbol{u}_r = \boldsymbol{r}_d^{(3)} + a_2 \ddot{\boldsymbol{e}}_r + a_1 \dot{\boldsymbol{e}}_r + a_0 \boldsymbol{e}_r \tag{5}$$

drives the tracking error $\boldsymbol{e}_r = \boldsymbol{r} - \boldsymbol{r}_d$ to converge to zero asymptotically, where $a_i > 0, i = 0, 1, 2$. For balance task, the roll motion is enforced to move along the balance equilibrium manifold under (5) as $\mathcal{E} = \{\varphi_b^e : \; f_1(\varphi_b^e) + h(\varphi_b^e) u_\psi = 0\}$. The balance control is updated as

$$\bar{u}_\psi = h^{-1}(\varphi_b)\left[J_t u_b - f_1(\varphi_b)\right], \; u_b = \ddot{\varphi}_b^e + b_1 e_b + b_0 e_b \tag{6}$$

where $b_0, b_1 > 0$ and $e_b = \varphi_b - \varphi_b^e$. The final control design incorporates both the above trajectory tracking and balance control, that is,

$$\boldsymbol{u} = [u_v \; \bar{u}_\psi]^T. \tag{7}$$

Under (7), the closed-loop system errors for both the trajectory tracking and roll balance converge to a small ball around zero exponentially [4], [10].

2) Assistive Leg Kinematics: The two 3-DOF legs are attached at S on the bikebot, see Fig. 1(a) and (b). The position of S in $\mathcal{B}$ is denoted as $\boldsymbol{r}_S = [l_b \; 0 \; h_b]^T$. The jth joint angle and link length are denoted as $\theta_{i,j}$ and $l_{i,j}$, $j = 0, 1, 2$, $i = L, R$, for left and right legs, respectively. The position of the first joint relative to the S is denoted as $[l_x \; l_y \; 0]^T$. We model the leg kinematics in $\mathcal{H}$. For the left and right legs, the homogeneous transformation matrices are then given by

$$\mathcal{A}_i^{\mathcal{H}} = \mathcal{A}_{\mathcal{B}}^{\mathcal{H}}(\varphi_b)\, \mathcal{A}_i^{\mathcal{B}}(\theta_i), \; i = L, R \tag{8}$$

where $\theta_i = [\theta_{i0} \; \theta_{i1} \; \theta_{i2}]^T$, $\mathcal{A}_{\mathcal{B}}^{\mathcal{H}} \in \mathbb{R}^{4\times4}$ is the homogeneous transformation matrix from $\mathcal{B}$ to $\mathcal{H}$, and $\mathcal{A}_i^{\mathcal{B}}$ describes the leg foot pose in $\mathcal{B}$. For the left foot, the position vector is $\boldsymbol{r}_L = [r_x \; r_y \; r_z]^T$ in $\mathcal{H}$ with

$$r_x = l_x - l_b + l_2 \, \mathrm{s}_{\theta_1+\theta_2} + l_1 \, \mathrm{s}_{\theta_1} \tag{9a}$$

$$r_y = \mathrm{c}_{\varphi_b}(l_y + l_0 \, \mathrm{c}_{\theta_0} - l_1 \, \mathrm{s}_{\theta_0} \, \mathrm{c}_{\theta_1} - l_2 \, \mathrm{s}_{\theta_0} \, \mathrm{c}_{\theta_1+\theta_2}) - \mathrm{s}_{\varphi_b}(h_b + l_0 \, \mathrm{s}_{\theta_0} + l_1 \, \mathrm{c}_{\theta_0} \, \mathrm{c}_{\theta_1} + l_2 \, \mathrm{c}_{\theta_0} \, \mathrm{c}_{\theta_1+\theta_2}) \tag{9b}$$

$$r_z = -\mathrm{c}_{\varphi_b}(h_b + l_0 \, \mathrm{s}_{\theta_0} + l_1 \, \mathrm{c}_{\theta_0} \, \mathrm{c}_{\theta_1} + l_2 \, \mathrm{c}_{\theta_0} \, \mathrm{c}_{\theta_1+\theta_2}) - \mathrm{s}_{\varphi_b}(l_y + l_{\theta_0} \, \mathrm{c}_{\theta_0} - l_1 \, \mathrm{c}_{\theta_1} \, \mathrm{s}_{\theta_0} - l_2 \, \mathrm{s}_{\theta_0} \, \mathrm{c}_{\theta_1+\theta_2}). \tag{9c}$$

Similar results are obtained for the right foot $\boldsymbol{r}_R$.

III. Wheel–Obstacle Impact Estimation

A. Wheel–Obstacle Impact Model

Fig. 1(c) shows the wheel–obstacle interactions. The front wheel hits the obstacle at point C. The wheel–obstacle interaction is considered as an elastic impact. The step-like obstacle has height h_o and the impact angle is denoted as $\phi_c = \phi_g + \psi$, where $\phi_g = \tan^{-1}(\frac{\mathrm{c}_\varepsilon}{\mathrm{c}_{\varphi_b}}\tan\phi)$ is the projected steering angle (along the front wheel plane). Denoting $\boldsymbol{q} = [x \; y \; z \; \psi \; \varphi_b]^T$, the impact model is written as

$$\boldsymbol{D}(\dot{\boldsymbol{q}}_f^+ - \dot{\boldsymbol{q}}_f^-) = \boldsymbol{I}_{\mathbb{I}}, \; \boldsymbol{D} = \begin{bmatrix} m_b \boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}_b \end{bmatrix} \tag{10}$$

where $\boldsymbol{I}_3$ is 3×3 identity matrix, $\boldsymbol{J}_b = \mathrm{diag}(J_z, J_t)$, and J_z is the moment of inertia matrix of the bikebot around the z-axis at G. Superscripts "+" and "−" in (10) represent the variables immediately after and before the impact and subscript "f" indicates the wheel–obstacle impact. The impulsive torque $\boldsymbol{I}_{\mathbb{I}}$ in (10) is related with the impact force $\boldsymbol{f}_{\mathbb{I}}$ at C as

$$\boldsymbol{I}_{\mathbb{I}} = \boldsymbol{J}_C^T \boldsymbol{f}_{\mathbb{I}} \tag{11}$$

where $\boldsymbol{J}_C = \frac{\partial \boldsymbol{r}_C(\boldsymbol{q})}{\partial \boldsymbol{q}} \in \mathbb{R}^{3\times5}$ is the Jacobian matrix, $\boldsymbol{r}_C(\boldsymbol{q}) = [x + l \, \mathrm{c}_\psi + L \, \mathrm{c}_{\psi_c} \; y + l \, \mathrm{s}_\psi + L \, \mathrm{s}_{\psi_c} \; z + h_o]^T$, $L = \sqrt{2R_w h_o - h_o^2}$, and R_w denotes the wheel radius.

Since $\boldsymbol{I}_{\mathbb{I}}$ in (10) is unknown, additional equations are needed to obtain $\dot{\boldsymbol{q}}_f^+$. Given an elastic impact at $t = t_f$, we denote $\dot{\boldsymbol{r}}_C|_{t=t_f^+} = \boldsymbol{J}_C \dot{\boldsymbol{q}}_f^+ = \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is the elastic velocity after impact and obtained by experiments. Combining (10) and (11), we obtain

$$\underbrace{\begin{bmatrix} \boldsymbol{D} & -\boldsymbol{J}_C^T \\ \boldsymbol{J}_C & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{D}_f} \begin{bmatrix} \dot{\boldsymbol{q}}_f^+ \\ \boldsymbol{f}_{\mathbb{I}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}\dot{\boldsymbol{q}}_f^- \\ \boldsymbol{\epsilon} \end{bmatrix}. \tag{12}$$

Because of full rank of $\boldsymbol{D}_f$, we obtain $\dot{\boldsymbol{q}}_f^+$ by inverting (12).

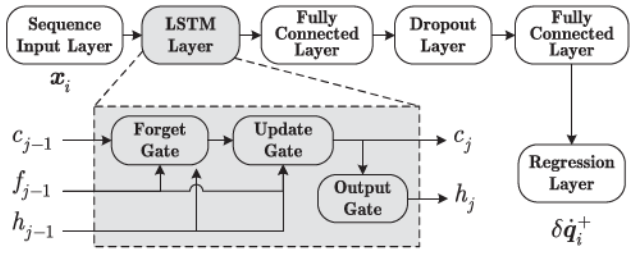

Fig. 2. Schematics of the LSTM-based estimation enhancement scheme.

B. Long Short-Term Memory (LSTM)-Based Impact Model

To account for model uncertainties and modeling errors, we extend (12) to use a machine learning method to improve the estimation. We consider the enhanced post-impact velocity estimation

$$\dot{\boldsymbol{q}}_f^* = \dot{\boldsymbol{q}}_f^+ + \delta\dot{\boldsymbol{q}}^+ \tag{13}$$

where $\dot{\boldsymbol{q}}_f^+$ is obtained from (12), and $\delta\dot{\boldsymbol{q}}^+$ denotes the correction term. We use an LSTM network model to estimate $\delta\dot{\boldsymbol{q}}^+$. Fig. 2 shows the structures of the LSTM network, including the input layer, LSTM layers, fully connected layers, and regression layer.

The neuron numbers at each layer are different, and multiple layers are used to build a deep neural network. The network builds the relationship between the bikebot motion measurements and the velocity correction $\delta\dot{\boldsymbol{q}}^+$. The LSTM network model is considered primarily because of its properties to capture nonlinear temporal relationship for real-time applications. We use the LSTM model to estimate analytical modeling errors and meanwhile, the analytical model is still taken to reduce the computational load and data-collection effort in the learning approach.

The LSTM network is trained offline. The training data are denoted as $\mathbb{D} = (\{\boldsymbol{x}_i\}_{i=1}^M, \{\delta\dot{\boldsymbol{q}}_i^+\}_{i=1}^M)$, where M is the number of data points, and $\boldsymbol{x}_i = \{\boldsymbol{a}_j, \boldsymbol{\omega}_j, v_j, \varphi_{bj}, \phi_j, \psi_j\}_{j=i-N+1}^i$ contains $N = 10$ observations of the IMU's accelerometer and gyroscope measurements (i.e., $\boldsymbol{a}_j \in \mathbb{R}^3$ and $\boldsymbol{\omega}_j \in \mathbb{R}^3$) and the bikebot motion and steering angle (i.e., v, φ_b, ϕ, ψ). The training data $\delta\dot{\boldsymbol{q}}_i^+$ are obtained from the difference of bikebot motion from the motion capture system and the impact model (12). In online prediction phase, the input variable $\boldsymbol{x}$ at the impact is fed into the trained LSTM network to predict $\delta\dot{\boldsymbol{q}}^+$. We will show the experimental validations in Section V.

IV. Assistive Impulsive Control Design

In this section, we first present the leg impulse control design and then discuss the integration with the EIC-based control.

A. Impulse Control Design

Denoting the interaction force as $\boldsymbol{F}$, the applied torque on the bikebot due to leg–terrain interaction is $\boldsymbol{\delta\tau} = \boldsymbol{r}_i \times \boldsymbol{F}$, where $\boldsymbol{r}_i$, $i = L, R$, is from (9). Given the left (right) leg joint torque vector $\boldsymbol{\tau}_\theta \in \mathbb{R}^3$, force $\boldsymbol{F}$ is calculated by $\boldsymbol{F} = (\boldsymbol{J}_\theta^T (\boldsymbol{R}_{\mathcal{H}}^{\mathcal{B}})^T)^{-1}\boldsymbol{\tau}_\theta$, where $\boldsymbol{R}_{\mathcal{B}}^{\mathcal{H}}$ is the rotation matrix from $\mathcal{B}$ to $\mathcal{H}$ and $\boldsymbol{J}_\theta$ is the Jacobian matrix of the foot. Therefore, we obtain the applied torque

$$\boldsymbol{\delta\tau} = \boldsymbol{r}_i \times \boldsymbol{F} = \boldsymbol{r}_i \times \left(\boldsymbol{J}_\theta^T \left(\boldsymbol{R}_{\mathcal{H}}^{\mathcal{B}}\right)^T\right)^{-1} \boldsymbol{\tau}_\theta. \tag{14}$$

The impulsive balance torque is the component of $\boldsymbol{\delta}_\tau$ along the x-axis direction, i.e., $\delta\tau_x = \boldsymbol{\delta\tau} \cdot \boldsymbol{e}_x$, $\boldsymbol{e}_x = [1\ 0\ 0]^T$. For balance purpose, we mainly focus on the control design of $\delta\tau_x$, and the torque components in the y-axis ($\delta\tau_y$) and z-axis ($\delta\tau_z$) directions are undesirable and, therefore, should be minimized.

We consider that impulsive torque $\boldsymbol{\delta\tau}$ is applied at $t = t_\tau$ with a short duration κ, that is, from t_τ to $t_\tau^+ := t_\tau + \kappa$. Under $\delta\tau_x$, the bikebot roll velocity experiences a discontinuous jump, while the roll angle remains the same. Using (3), we obtain

$$\lim_{\kappa\to 0}\int_{t_\tau}^{t_\tau^+} J_t\ddot{\varphi}_b dt = \lim_{\kappa\to 0}\int_{t_\tau}^{t_\tau^+} [f_1(\varphi_b) + h(\varphi_b)u_\psi + \delta\tau_x]dt.$$

Note that $\lim_{\kappa\to 0}\int_{t_\tau}^{t_\tau^+}[f_1(\varphi_b) + h(\varphi_b)u_\psi]dt = 0$ since the roll angle remains unchanged during $[t_\tau, t_\tau^+]$ under an ideal impulse. Therefore, from the above equation, we obtain

$$J_t\Delta\dot{\varphi}_b = \tau_{\mathbb{I}} := \int_{t_\tau}^{t_\tau^+} \delta\tau_x dt \tag{15}$$

where $\Delta\dot{\varphi}_b = \dot{\varphi}_{b\tau}^+ - \dot{\varphi}_{b\tau}^-$ is the roll angular velocity change, and $\varphi_{b\tau}^- = \varphi_b(t_\tau)$ and $\varphi_{b\tau}^+ = \varphi_b(t_\tau^+)$ represent the roll angles before and after $\delta\tau_x$ is applied, respectively.

From (12) and (15), it is clear that designing $\delta\tau_x$ (i.e., $\tau_{\mathbb{I}}$) is equivalent to identify the desired roll angular rate $\dot{\varphi}_{b\tau}^+$ and linear velocity v_τ^+. We, thus, formulate the following optimization problem:

$$\min_{\dot{\varphi}_{b\tau}^+, v_\tau^+} \int_{t_\tau^+}^{t_\tau^+ + H_t} \left(\boldsymbol{x}_e^T \boldsymbol{P} \boldsymbol{x}_e + \boldsymbol{u}^T \boldsymbol{Q} \boldsymbol{u}\right) dt \tag{16a}$$

$$\text{subject to: (1), (3), (7) , and (15)} \tag{16b}$$

$$|\delta\tau_x| \le \delta\tau_x^{\max}, |v_\tau^+| \le v^{\max}, |\phi| \le \phi^{\max}, \tag{16c}$$

where $\boldsymbol{x}_e = [\boldsymbol{e}_r^T\ \dot{\boldsymbol{e}}_r^T\ e_b\ \dot{e}_b]^T$ is the error vector, matrices $\boldsymbol{P} \in \mathbb{R}^{6\times 6}$ and $\boldsymbol{Q} \in \mathbb{R}^{2\times 2}$ are symmetric positive definite, and $H_t > 0$ is the predictive horizon. Condition (16a) denotes the tracking and balance errors and control efforts over the horizon H_t, and (16c) includes the physical limits. The optimization problem (16) is in the form similar to the model predictive control (MPC) formulation [22]. The calculation of errors $\boldsymbol{e}_r$ and e_b requires to compute the predictive state variables of systems dynamics given by (1), (3), and (15) over horizon H_t under control $\boldsymbol{u}$ given in (7), that is, these system dynamics are integrated over $t \in [t_\tau^+, t_\tau^+ + H_t]$ to obtain the predictive states $\boldsymbol{r}(t)$ and $\varphi_b(t)$, $t_\tau^+ \le t \le t_\tau^+ + H_t$. Unlike MPC, instead of searching the control policy, the optimization problem in (16) is formulated to find the optimal initial velocities ($\dot{\varphi}_{b\tau}^+$ and v_τ^+) of the bikebot system to restart the trajectory tracking. We solve (16) using the sequential quadratic programming algorithm [23].

The value of κ is selected as small as possible to keep leg–terrain contact duration short. We use constant $\delta\tau_x$, and from (15), we have

$$\delta\tau_x = \frac{1}{\kappa} J_t \left(\dot{\varphi}_{b\tau}^+ - \dot{\varphi}_{b\tau}^-\right) = \frac{1}{\kappa} J_t \Delta\dot{\varphi}_b,\ t \in [t_\tau, t_\tau^+). \tag{17}$$

To generate $\delta\tau_x$ given in (17), we need to design the leg–terrain interaction force $\boldsymbol{F}$. The terrain condition where the leg touches down for impulse generation can be obtained by using onboard sensors (e.g., camera or LiDAR) and therefore, is assumed to be known. When leg touches the terrain, contact point is located at $\boldsymbol{r}_i = [r_{ix}\ r_{iy}\ 0]^T$, and $\delta\tau_x$ is then expressed as $\delta\tau_x = r_{iy}F_z$, where $F_z = \boldsymbol{F} \cdot \boldsymbol{e}_z$, $\boldsymbol{e}_z = [0\ 0\ 1]^T$. Meanwhile, to eliminate the possible effects by $\delta\tau_y$ and $\delta\tau_z$, we enforce constraints $F_x = F_y = 0$ and $r_{ix} = 0$. In practice, the wheel–obstacle impact often prevents the bikebot platform to rotate in the y- and z-axis directions, and therefore, the magnitudes of $\delta\tau_y$ and $\delta\tau_z$ are small and can be negligible. Given r_{iy}, the force F_z then is determined as

$$F_z = \frac{\delta\tau_x}{r_{iy}},\ F_z \le F_z^{\max} \tag{18}$$

where $F_z^{\max}$ is the maximum vertical ground reaction force. The leg–terrain contact point position r_{iy} is offline tuned and adjusted in experiment. The joint angles $\boldsymbol{\theta}$ are calculated by inverse kinematics (9), and the joint torques are obtained as

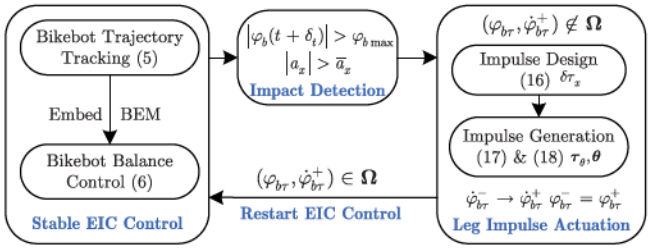

Fig. 3. Illustrative schematic of the integrated EIC-based control and assistive impulse control.

$\boldsymbol{\tau}_\theta = \boldsymbol{J}_\theta^T (\boldsymbol{R}_{\mathcal{H}}^{\mathcal{B}})^T \boldsymbol{F}$ with $\theta_j^{\min} \le \theta_{i,j} \le \theta_j^{\max}$, $|\tau_{\theta_{i,j}}| \le \tau_\theta^{\max}$, $i = L, R$, $j = 1, 2, 3$.

B. Controller Integration and Analysis

The EIC-based control is continuous, while the impulse control has a discrete feature. Integration of those two control algorithms is, thus, a hybrid control strategy. Fig. 3 shows the integration design of the two controllers.

The leg impulse control is triggered when the wheel hits the obstacle and the roll angle then reaches $\varphi_b^{\max}$, the maximum controllable roll angle under the EIC-based control. Note that at this time, the roll motion state $(\varphi_{b\tau}^-, \dot{\varphi}_{b\tau}^-)$ is out of the region of attraction, denoted by Ω, of the closed-loop roll motion dynamics under the EIC-based control, that is, $(\varphi_{b\tau}^-, \dot{\varphi}_{b\tau}^-) \notin \Omega$. The real-time wheel–terrain impact is detected through the acceleration measurement $|a_x|$ greater than a predefined threshold $a_x^{\max}$. The estimate in (13) is used to obtain the state $\dot{\boldsymbol{q}}_f^*$ right after wheel–terrain impact. The roll motion is then predicted by (4) for a horizon δ_t at t under the EIC-based control. If $\varphi_b(t) > \varphi_b^{\max}$ in $[t, t+\delta_t)$, the leg–terrain impulsive control is initiated. In this case, (16) and (17) are used to design the impulsive torque $\delta\tau_x$, and (18) is used to obtain the leg poses θ and torques $\boldsymbol{\tau}_\theta$. Since the leg impulse control only helps balance task, the trajectory tracking still needs to be maintained by the EIC-based control given by (7).

We analyze the bikebot roll motion under the impulse control. The roll dynamics for $t \in [t_\tau, t_\tau^+)$ is obtained as

$$\dot{\varphi}_b(t) = \dot{\varphi}_{b\tau}^- + \frac{1}{J_t}\int_{t_\tau}^{t} \delta\tau_x dt = \dot{\varphi}_{b\tau}^- + \frac{t - t_\tau}{\kappa}\Delta\dot{\varphi}_b. \tag{19}$$

It is clear that $\dot{\varphi}_b(t_\tau^+) = \dot{\varphi}_{b\tau}^+$ and integrating (19) leads to

$$\varphi_b(t) = \varphi_b(t_\tau) + \frac{\Delta\dot{\varphi}_b}{2\kappa}\left(t^2 - t_\tau^2 - 2\kappa t_\tau\right). \tag{20}$$

Because of $\kappa \ll 1$, from (20), we obtain that $\varphi_b(t_\tau^+) = \varphi_{b\tau}^+ \approx \varphi_b(t_\tau) = \varphi_{b\tau}^-$, that is, $\delta\tau_x$ in (15) is nearly an ideal impulse.

For roll motion at $t \ge t_\tau^+$, we simplify the steering torque $\tau_s = -m_b h_G \frac{v^2}{l} c_\epsilon \tan\phi$, and (4) is linearized as

$$\ddot{\varphi}_b - k_1^2 \varphi_b + k_2 \tan\phi = 0 \tag{21}$$

where $k_1^2 = \frac{m_b h_G g}{J_t}$ and $k_2 = \frac{m_b h_G v^2}{J_t l}$. Letting $\varsigma = t - t_\tau^+$ and $k_3 = k_2/k_1$, the solution to (21) is then obtained as

$$\varphi_b(\varsigma) = c_1(\varsigma)\varphi_{b\tau}^+ + c_2(\varsigma)\dot{\varphi}_{b\tau}^+ + k_3\int_0^\varsigma c_2(\varsigma - s)\tan\phi(s)ds$$

where $c_1(\varsigma) = \cosh(k_1\varsigma)$ and $c_2(\varsigma) = \sinh(k_1\varsigma)$. Substituting (17) into the above equation, we obtain

$$\varphi_b(\varsigma) = c_1(\varsigma)\varphi_{b\tau}^- + c_2(\varsigma)\dot{\varphi}_{b\tau}^- + k_3\int_0^\varsigma c_2(\varsigma - s)\tan\phi(s)ds + c_2(\varsigma)\kappa J_t \delta\tau_x.$$

When state $(\varphi_{b\tau}^-, \dot{\varphi}_{b\tau}^-) \notin \Omega$, they have the same sign (see Fig. 9). In this case, the steering angle $\phi(t)$ needs to reach the maximum value $\phi_{\max}$ for balance control, and we then approximate roll angle as

$$\varphi_b(\varsigma) \approx c_1(\varsigma)\varphi_{b\tau}^- + c_2(\varsigma)\dot{\varphi}_{b\tau}^- + k_3 c_1(\varsigma)\tan\phi_{\max} + c_2(\varsigma)\kappa J_t\delta\tau_x.$$

We collect all terms related to $e^{k_1\varsigma}$ in the above equation, and to prevent the divergence, we obtain the necessary condition

$$\operatorname{sgn}(\dot{\varphi}_{b\tau}^-)\left(\varphi_{b\tau}^- + \dot{\varphi}_{b\tau}^- + k_3\tan\phi_{\max} + \kappa J_t\delta\tau_x\right) < 0. \tag{22}$$

If $\dot{\varphi}_{b\tau}^- > 0$, the applied impulsive torque $\delta\tau_x$ is negative and the right leg actuation is initiated. Similarly, if $\dot{\varphi}_{b\tau}^- < 0$, $\delta\tau_x$ is positive and the left leg is used in actuation. From (22), we obtain the condition for $\delta\tau_x$ as

$$|\delta\tau_x| > \delta\tau_x^{\min} := \frac{|\varphi_{b\tau}^- + \dot{\varphi}_{b\tau}^- + k_3\tan\phi_{\max}|}{\kappa J_t}. \tag{23}$$

Condition (23) implies that under the EIC-based control (7), when the bikebot is about to lose balance, i.e., $(\varphi_{b\tau}^-, \dot{\varphi}_{b\tau}^-) \notin \Omega$, the impulsive torque magnitude should be greater than $\delta\tau_x^{\min}$ to bring $(\varphi_{b\tau}^+, \dot{\varphi}_{b\tau}^+) \in \Omega$ to recover the balance.

The stability of the integrated EIC-based motion and leg impulse control lies in the controller design mechanism, as shown in Fig. 3. The switching condition from the EIC-based control is triggered by the event when the bikebot hits on obstacles and bikebot roll motion $(\varphi_{b\tau}, \varphi_{b\tau}^-) \notin \Omega$. The leg–terrain impulse control is then designed such that under the impulse actuation, the roll motion velocity changes immediately (roll angles are kept the same, $\varphi_{b\tau}^- = \varphi_{b\tau}^+$) and the new state $(\varphi_{b\tau}, \varphi_{b\tau}^+)$ is within Ω, that is, $(\varphi_{b\tau}, \varphi_{b\tau}^+) \in \Omega$. This is achieved by using the optimization design for $\varphi_{b\tau}^+$ (and v_τ^+) by (16) and impulse generation by (17) and (18). The stability of the overall system is, therefore, guaranteed.

V. Experiments

A. Experimental Setup

Fig. 1(a) shows the bikebot prototype, and Fig. 4 shows the embedded system design. The bikebot velocity was measured by an encoder (200 pulses per revolution), and an IMU (angular measurement precision around 0.05°) was mounted on the frame to measure the roll and yaw motion. The onboard embedded system includes a low-level real-time microcontroller (model STM32F429) and the upper level high-performance platform (Nvidia Jetson TX2) that runs robot operating system (ROS). The leg impulse control was implemented on the ROS-based platform through a CAN bus interface. The leg control and the EIC-based control were implemented at the frequencies of 1000 and 50 Hz, respectively. No force sensor was used to measure the leg–terrain interaction force, and instead, the leg joint torques were measured in real time to estimate the interaction forces for

TABLE I
PHYSICAL PARAMETERS OF THE BIKEBOT AND THE ASSISTIVE LEGS

m_b (kg)	J_b (kgm^2)	J_z (kgm^2)	l (m)	R_w (m)	l_G/h_G (m)	l_b/h_b (m)	ε (°)	l_j (m)	l_x, l_y (m)	$\tau_{\theta_j}^{\max}$ (Nm)	$\theta_j^{\max}/\theta_j^{\min}$ (°)
24	0.25	0.5	0.87	0.225	−0.04/0.35	−0.14/0.26	17	0.075/0.212/0.207	0.1/0.13	25	150/ − 150

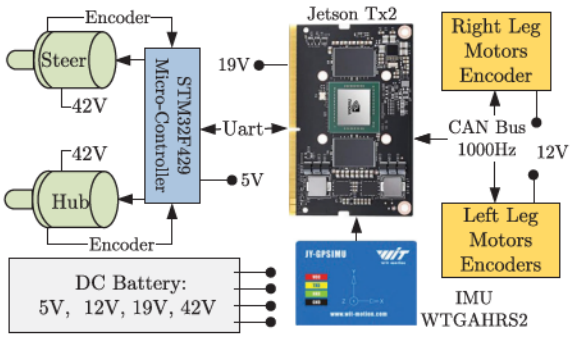

Fig. 4. Schematic of the embedded system for the bikebot.

Fig. 5. Bikebot experiments for various obstacle crossing tests. Wooden blocks with different heights are used as obstacles in experiments.

impulse control. We conducted experiments and validated such an approach by using an external 6-DOF force sensor (model M3703 C from Sunrise Instruments, Nanning, China).

Fig. 5 shows the experimental setup. We used a set of wooden blocks to construct obstacles with different heights. Irregular obstacles were also constructed using these wooden blocks as well as other objects (e.g., mechanical structures, steel water bottle). Table I lists the values of the physical model parameters and the leg joint angle and torque range limits. In experiments, the impulsive duration was selected as $\kappa = 50$ ms, and the prediction horizon in (16) was chosen $H_t = 1$ s. Other optimization and controller parameters values include: $\delta_t = 200$ ms, $\delta\tau_x^{\max} = 30$ N · m, $F_z^{\max} = 190$ N, $\phi^{\max} = 30°$, $v^{\max} = 1.5$ m/s, $a_x^{\max} = 5$ m/s^2, $\varphi_b^{\max} = 5°$, $\boldsymbol{P} = \text{diag}(1, 1, 1, 1, 10, 10)$, and $\boldsymbol{Q} = \text{diag}(10, 10)$. The EIC-based control parameters were chosen as $a_1 = 3$, $a_2 = 6$, $a_3 = 10$, $b_1 = 180$, and $b_2 = 25$. The abovementioned force and toque bounds were mainly determined by the leg joint motor torque limits and from static leg-firm ground contact experiments. The control parameters selected previously were tested through simulations and preliminary experiments. We tuned the EIC-based control gains such that the bikebot balance was always fulfilled, even when the trajectory tracking performance was not satisfactory. For matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$, the large elements were used for the roll motion error and the penalizing weights between the tracking errors and control efforts, respectively.

B. Experimental Results

We first present the leg force control and impact model validation results. Fig. 6(a) shows the leg force control validation. A leg–terrain interaction force $F_z = 30$ N with duration 1.5 s was commanded. The leg–terrain interaction force was calculated by the joint torque measurements and the model given in (14). The model calculation was compared with the ground truth measured by the external force sensor. It is clear that the joint torque feedback can be used to maintain the force control as the results match the measurements. Fig. 6(b) further shows the impact model prediction results. The LSTM was trained by the dataset with over $M = 10\,000$ points. These data were collected in experiments to run across obstacles with various heights. The ground truth information of the bikebot motion was obtained from an optical motion capture system. The roll and velocity estimations by the impact model (12) were not in good agreement with the ground truth, while with the LSTM-based enhancement, the results closely matched the ground truth. Fig. 6(c) further shows the prediction errors by the LSTM-enhanced results from multiple experiments, which consistently confirm small errors. The LSTM prediction running time was around 16 ms on average and can be implemented in real time.

Fig. 7 shows the impulse control results in a straight-line tracking experiment. A rectangular wooden block with height $h_o = 7.8$ cm was used as the obstacle. Fig. 7(a) and (b) shows the bikebot trajectory and velocity with and without the impulse input. At $t = 2.3$ s, the wheel hit the obstacle and the velocity significantly dropped for about 1 m/s. In Fig. 7(c), roll angle increased monotonically to $\varphi_b^{\max} = 5°$, and only with the steering control, the bikebot cannot be kept balance. When the impulse control was triggered at around $t_\tau = 3.05$ s, the right leg hit the ground to generate impulsive torque $\delta\tau_x = 20$ N · m, see Fig. 7(e). Roll velocity changed from $\dot{\varphi}_{b\tau}^- = 20$ to $\dot{\varphi}_{b\tau}^+ = -40°$/s; see Fig. 7(d). Under this impulse control, roll angle moved back to around zero and the bikebot motion was back into the region of attraction Ω under the EIC-based control with an increased speed $v = 1.1$ m/s at $t = 4$ s. The result without the impulse control showed the loss of balance (with roll angle over 10°). Fig. 7(e) and (f) shows the right leg joint angles and torques, impulsive torques, and the leg–terrain interaction forces. Within the impulse duration, roll angle change was small (around 0.3°). Foot contact angle $\theta_0 \approx 0$ implies that the leg link touched the ground nearly vertically. Under this configuration, vertical contact force F_z played a major role for impulsive torque generation. These results confirmed the effectiveness of the impulsive torque control design.

We further demonstrate the control performance of the bikebot running across obstacles with different heights. Fig. 8(a) shows

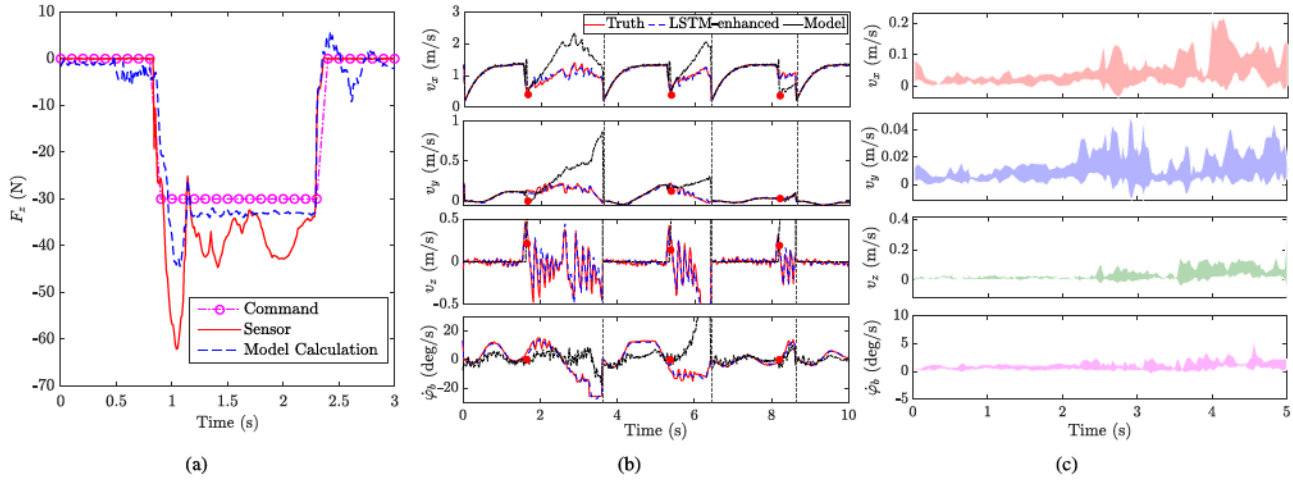

Fig. 6. (a) Validation experiment of the leg–terrain interaction force control by using the join torque feedback. (b) Wheel–obstacle impact model estimation results by the impact model and the LSTM-based enhancement. The red circular markers " • " denote the q_f^+ estimates by the impact model. The vertical lines indicates different experimental trials. (c) Prediction errors of multiple trials with mean and standard deviation. The time duration is normalized to 5 s.

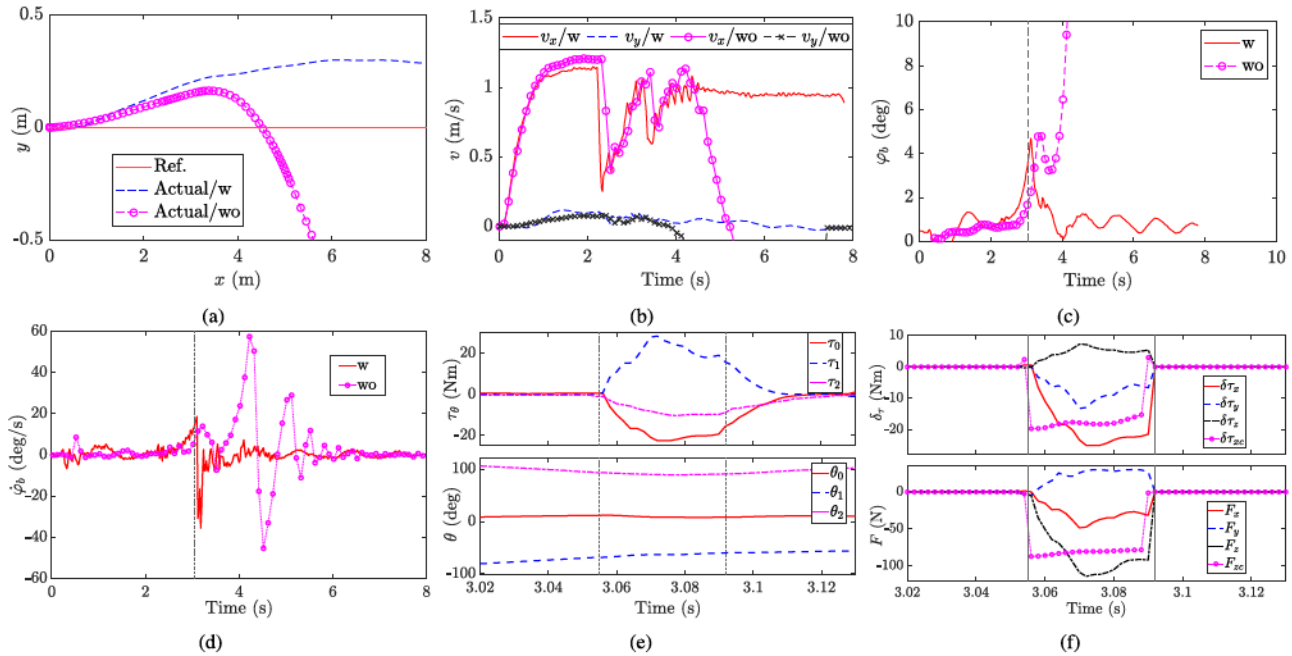

Fig. 7. Comparison results with and without leg assistive impulse control for a straight-line trackin experiments. (a) Bikebot position in the planar plane. (b) Bikebot velocities v_x and v_y. (c) Bikebot roll angle φ_b. (d) Bikebot roll angular velocity $\dot{\varphi}_b$. (e) Leg joint torque and joint angle. (f) Leg–terrain interaction force and torque. The vertical lines in (c)–(f) indicate impulse control time.

the results of the bikebot crossing four obstacles separately at heights $h_o = 2$ cm (Case 1), 4 cm (Case 2), 5 cm (Case 3), and 6 cm (Case 4). For a thin obstacle (Case 1), the EIC-based control successfully maintained the balance without need of the leg impulse control, showing the robustness of the EIC-based control design. For others three cases, the wheel–obstacle impacts were significant due to large obstacle heights and the leg impulse control was needed. To run crossing obstacles with large heights, roll velocity changes were large and significant impulsive torques were needed, see the bottom plot in Fig. 8(a). We further conducted experiments to run across three obstacles continuously with heights $h_o = 3.5$, 6, and 5 cm. Fig. 8(b) and (c) shows the roll motion and the leg assistive impulsive torques. The bikebot successfully ran across the first obstacle by the EIC-based control, and for the last two obstacles, the impulse control was triggered to regain the balance. Roll angles moved back to around zero in a short duration, and the left and right legs were sequentially used to generate the impulsive torques, as shown in Fig. 8(c). We conducted additional experiments under manually pushing disturbances on the bikebot and crossing

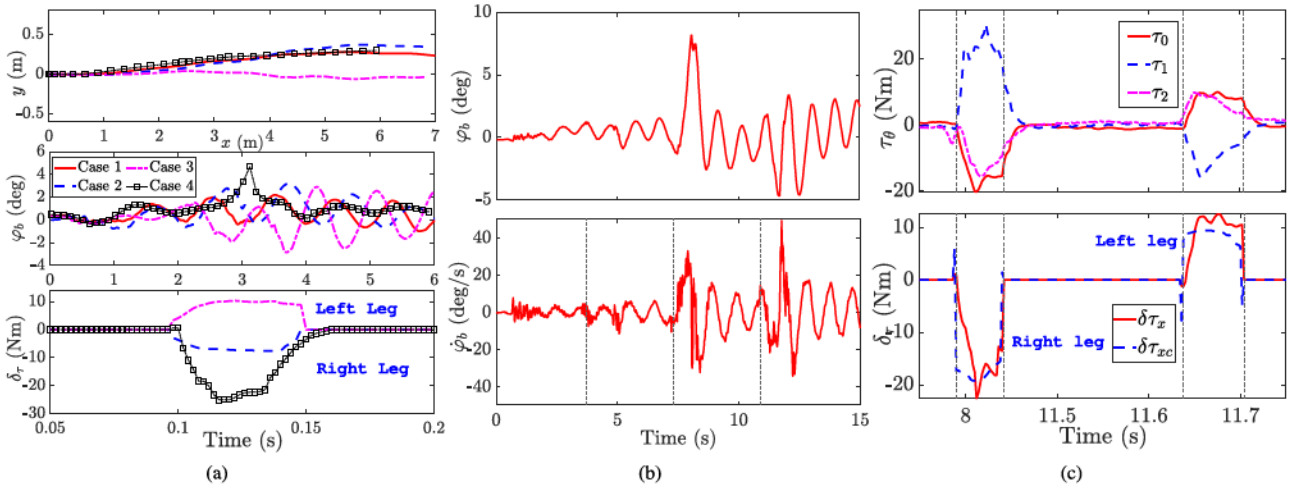

Fig. 8. (a) Experimental results for straight-line tracking and crossing one obstacle with different heights. Case 1: $h_o = 2$ cm. Case 2: $h_o = 4$ cm. Case 3: $h_o = 5$ cm. Case 4: $h_o = 6$ cm. Top: Planar trajectory; middle: roll angles; bottom: applied impulse torques. (b) and (c) are experimental results to run crossing three obstacles in one run at heights $h_o = 3.5, 6, 5$ cm continuously. (b) Bikebot roll motion. The wheel–obstacle impacts happen at the moments indicated by the vertical lines. (c) Leg joint torques and impulse torques. The positive and negative $\delta\tau_x$ are applied by the left and right legs, respectively.

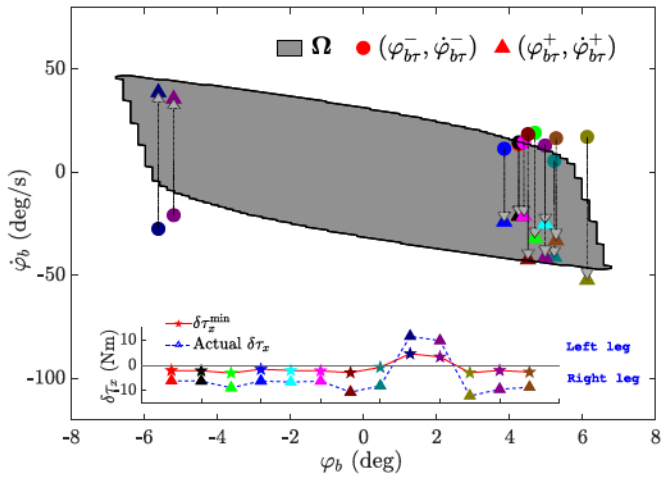

Fig. 9. Phase portrait of the preimpulse and postimpulse states and their relationship with the estimated region of attraction of the EIC-based control. Markers "•" and "▲" indicate the preimpulse state $(\varphi_{b\tau}^-, \dot{\varphi}_{b\tau}^-)$ and postimpulse state $(\varphi_{b\tau}^+, \dot{\varphi}_{b\tau}^+)$ in experiments, respectively.

irregular-shape obstacles. The results also confirmed that the leg impulse control was critical to help balance the bikebot under these unmodeled uncertainties and disturbances.

To show and validate stability under the integrated EIC-based motion and leg impulse control, Fig. 9 shows the relationship of the bikebot roll motion in the $(\varphi_b, \dot{\varphi}_b)$ plane. The shaded area shows the region of attraction Ω under the EIC-based controller, which was obtained by computation with imposing the same constraints in (16) for straight-line tracking task. We plotted the bikebot roll motion before (circular markers) and after (triangular markers) the impulse control in multiple experimental runs. It is clear that the pre-impulse state $(\varphi_{b\tau}^-, \dot{\varphi}_{b\tau}^-)$ are at the boundary or outside of Ω, while the post-impulse state $(\varphi_{b\tau}^+, \dot{\varphi}_{b\tau}^+)$ jumped and moved back into Ω. This implies that the designed impulse control guarantees the roll motion state back to Ω and reinitializes the EIC-based control to achieve stability. The bottom plot in Fig. 9 further shows that actual impulsive torques $\delta\tau_x$ in experiments satisfy the analytical bounds in (23). These results confirm the integrated impulse balance control design and the stability analysis.

VI. Conclusion

This article presented an integrated impulse control with nonlinear trajectory tracking and balance (i.e., EIC-based control) for a single-track autonomous bikebot to safely run across obstacles. Two assistive legs were attached to the bikebot to provide leg–terrain impulsive torques when crossing obstacles. The impulse control was built on an optimization-based approach, and a switching strategy was designed to integrate with the EIC-based controller. An LSTM-based model was used to improve estimation accuracy for the wheel–obstacle impact in real time. Experimental results demonstrated that with the proposed mechatronic and control design, the bikebot can navigate crossing step-like obstacles with heights more than one-third of the wheel radius. Extensive experiments were presented to demonstrate the robustness of the control systems design over various obstacles.

References

[1] A. Muetze and Y. C. Tan, "Electric bicycles—A performance evaluation," *IEEE Ind. Appl. Mag.*, vol. 13, no. 4, pp. 12–21, July/Aug. 2007.
[2] F. Han and J. Yi, "Stable learning-based tracking control of underactuated balance robots," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1543–1550, Apr. 2021.
[3] K. J. åström, R. E. Klein, and A. Lennartsson, "Bicycle dynamics and control," *IEEE Control Syst. Mag.*, vol. 25, no. 4, pp. 26–47, Aug. 2005.
[4] N. Getz, "Dynamic inversion of nonlinear maps with applications to nonlinear control and robotics," Ph.D. dissertation, Dept. Electr. Eng. Comp. Sci., Univ. Calif., Berkeley, CA, USA, 1995.

[5] J. Yi, D. Song, A. Levandowski, and S. Jayasuriya, "Trajectory tracking and balance stabilization control of autonomous motorcycles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 2583–2589.
[6] Y. Tanaka and T. Murakami, "A study on straight-line tracking and posture control in electric bicycle," *IEEE Trans. Ind. Electron.*, vol. 56, no. 1, pp. 159–168, Jan. 2009.
[7] L. Chen *et al.*, "Robust control of reaction wheel bicycle robot via adaptive integral terminal sliding mode," *Nonlinear Dyn.*, vol. 104, pp. 2291–2302, 2021.
[8] F. Han, A. Jelvani, J. Yi, and T. Liu, "Coordinated pose control of mobile manipulation with an unstable bikebot platform," *IEEE/ASME Trans. Mechatron.*, 2022, early access, Apr. 6, 2022, doi: 10.1109/TMECH.2022.3157787.
[9] Y. Zhang, "Modeling and control of single track vehicles: A human/machine/environment interactions perspective," Ph.D. dissertation, Dept. Mech. Aero. Eng., Rutgers Univ., Piscataway, NJ, USA, 2014.
[10] P. Wang, J. Yi, T. Liu, and Y. Zhang, "Trajectory tracking and balance control of an autonomous bikebot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 2414–2419.
[11] L. Cui, S. Wang, S. Yang, Z. Zhang, and Z.-P. Jiang, "Nonlinear balance control of an unmanned bicycle: Design and experiments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2020, pp. 7279–7284.
[12] Y. Zhang, P. Wang, J. Yi, D. Song, and T. Liu, "Stationary balance control of a bikebot," in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, 2014, pp. 6706–6711.
[13] Y. Kim, H. Kim, and J. Lee, "Stable control of the bicycle robot on a curved path by using a reaction wheel," *J. Mech. Sci. Technol*, vol. 29, no. 5, pp. 2219–2226, 2015.
[14] P. Seekhao, K. Tungpimolrut, and M. Parnichkun, "Development and control of a bicycle robot based on steering and pendulum balancing," *Mechatronics*, vol. 69, pp. 1–12, 2020.
[15] A. Shkolnik, M. Levashov, I. R. Manchester, and R. Tedrake, "Bounding on rough terrain with the LittleDog robot," *Int. J. Robot. Res.*, vol. 30, no. 2, pp. 192–215, 2011.
[16] G. Valsecchi, R. Grandia, and M. Hutter, "Quadrupedal locomotion on uneven terrain with sensorized feet," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1548–1555, Apr. 2020.
[17] H. Peng, J. Wang, S. Wang, W. Shen, D. Shi, and D. Liu, "Coordinated motion control for a wheel-leg robot with speed consensus strategy," *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 3, pp. 1366–1376, Jun. 2020.
[18] W.-H. Chen, H.-S. Lin, Y.-M. Lin, and P.-C. Lin, "Turboquad: A novel leg–wheel transformable robot with smooth and fast behavioral transitions," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1025–1040, Oct. 2017.
[19] N. Kant, R. Mukherjee, D. Chowdhury, and H. K. Khalil, "Estimation of the region of attraction of underactuated systems and its enlargement using impulsive inputs," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 618–632, Jun. 2019.
[20] S. Weibel, G. W. Howell, and J. Baillieul, "Control of single-degree-of-freedom hamiltonian systems with impulsive inputs," in *Proc. IEEE Conf. Decis. Control*, Kobe, Japan, 1996, pp. 4661–4666.
[21] P. Wang, J. Yi, and T. Liu, "Stability and control of a rider-bicycle system: Analysis and experiments," *IEEE Trans. Automat. Sci. Eng.*, vol. 17, no. 1, pp. 348–360, Jan. 2020.
[22] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Santa Barbara, CA, USA: Nob Hill Publishing, 2020.
[23] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 2006.