Coordinated Pose Control of Mobile Manipulation With an Unstable Bikebot Platform

Feng Han, *Student Member, IEEE*, Alborz Jelvani, Jingang Yi, *Senior Member, IEEE*, and Tao Liu, *Senior Member, IEEE*

***Abstract*—Bikebot manipulation has advantages of the single-track robot mobility and manipulation dexterity. We present a coordinated pose control of mobile manipulation with the stationary bikebot. The challenges of the bikebot manipulation include the limited steering balance capability of the unstable bikebot and kinematic redundancy of the manipulator. We first present the steering balance model to analyze and explore the maximum steering capability to balance the stationary platform. A balancing equilibrium manifold is then proposed to describe the necessary condition to fulfill simultaneous platform balance and posture control of the end-effector. A coordinated planning and control design is presented to determine the balance-prioritized posture control under kinematic and dynamic constraints. Extensive experiments are conducted to demonstrate the mechatronic design for autonomous plant inspection in agricultural applications. The results confirm the feasibility to use the bikebot manipulation for plant inspection with end-effector position and orientation errors about 5 mm and 0.3 degs, respectively.**

***Index Terms*—Balance control, bicycle control, mobile manipulation, task priority planning, underactuated robots.**

I. Introduction

MOBILE manipulation integrates a mobile robot with an onboard multilink manipulator to expand workspace and improve capability for complex manipulation tasks [1]–[3]. Mobile manipulation can be built on wheeled, legged, or aerial platforms and the applications include agriculture harvesting [4], mobile cranes [5], underwater archaeology [6], and aerial manipulation [7], [8], etc. The advantages of the mobile manipulation come at the cost of coordinated planning and control [9]. The coupled dynamics of the mobile platform and manipulator is one of the design challenges [10]. Unknown or uncertain robot-environment interactions bring additional complexity for control of mobile manipulation [5], [11]. For instance, a wheeled/legged mobile robot would fall down when moving on a steep or rocky field [12]. For aerial manipulation, interaction forces generates large disturbances for robot motion and control due to limited actuation of the quadrotors [7], [13].

Coordinated planning and control is critical when the mobile platform is unstable or in complex, dynamic environments. Balance control of unstable platform is among the highest priority tasks for mobile manipulation. In [14], a model predictive control was presented for collaborative manipulation, balancing, and interaction of a ball-based three degree-of-freedom (DOF) manipulator. In [15], a single spherical wheel-based "ballbot" was used as the platform for mobile manipulation and a balance motion control was developed for the underactuated, nonholonomic robot. For kinematic redundant manipulators, task-priority control takes advantage of design space in the null space of the Jacobian matrix. Optimization-based velocity control was designed to specify tasks from the highest to lowest priorities [16]–[18].

In this article, we present a mobile manipulation system that is built on a bikebot (i.e., autonomous bicycle). A six-DOF lightweight manipulator is mounted on the bikebot and the system was developed for agricultural applications, such as autonomous plant inspection and scouting [19]. All existing agricultural robots are built on double-track mobile platform and their energy consumption is much higher than that of single-track mobile robots, such as bikebot [20]. It is challenging for double-track robots to navigate in narrow, cluttered spaces and to actively probe and flexibly inspect objects under the canopy of densely grown, tall plants. Light-weight bikebot provides additional advantages for small footprints that potentially avoid potential severe soil compaction [21]. Steering and speed control of autonomous one-wheel-steered bikebot has been reported (e.g., [22], [23]), but balance control of two-wheel steered-bikebot for manipulation has not been studied. Because of the unstable platform and limited actuation, assistive devices were used to generate additional balance torque [24]–[27]. However, the additional balance actuators increase the systems complexity and operation cost. In this work, steering is used as the only actuation for balancing the stationary platform.

Manuscript received July 22, 2021; revised November 27, 2021 and January 29, 2022; accepted March 4, 2022. Recommended by Technical Editor and Senior Editor M. Basin. This work was supported in part by the U.S. National Science Foundation under Grant CNS-1932370(Yi), in part by the Natural Science Foundation of China under Grant 52175033 and Grant U21A20120, and in part by Zhejiang Provincial Natural Science Foundation under Grant LZ20E050002 (Liu). *(Corresponding author: Jingang Yi.)*

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).

Alborz Jelvani is with the Department of Computer Science, Rutgers University, Piscataway, NJ 08854 USA (e-mail: aj654@scarletmail.rutgers.edu).

Tao Liu is with the State Key Laboratory of Fluid Power and Mechatronic Systems and the School of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: liutao@zju.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMECH.2022.3157787.

Digital Object Identifier 10.1109/TMECH.2022.3157787

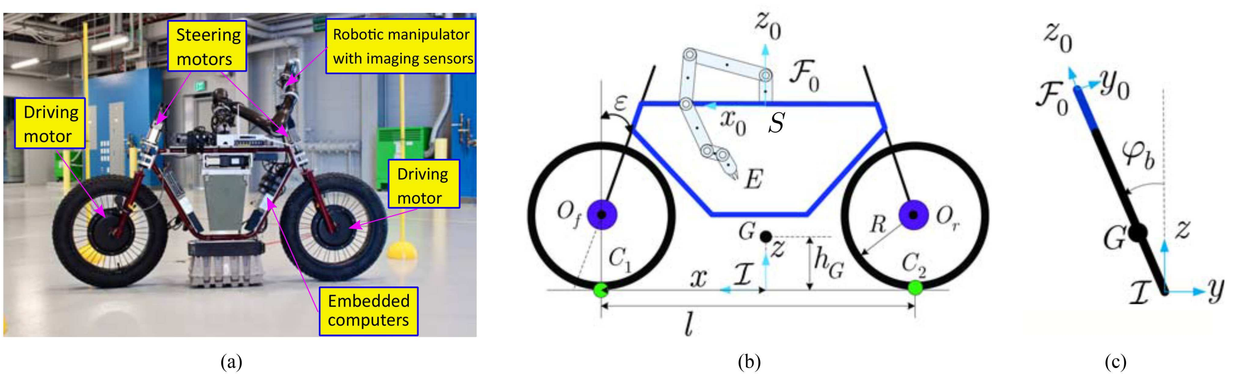


Fig. 1. (a) Prototype of the two-wheel steered bikebot mobile manipulation system. (b) Side-view configuration of the bikebot manipulation system. (c) Fron-view schematic of the bikebot roll motion.

We focus on stationary balance of the bikebot manipulation for several reasons. First, it is more challenging to balance a stationary bikebot than a moving platform. With front wheel steering actuation, the bikebot can be only balanced within a small range of roll motion at stationary (i.e., 2–3 degs) [28], [29]. It is desirable to design new mechanisms and control methods to enlarge the controllable roll motion range for practical applications. Second, many applications, such as plant inspection require that the mobile platform stays stationary, while the onboard manipulator conducts the visual inspection or sample manipulation tasks. Therefore, robotic applications require the stationary balance capability.

This article presents the coordinated control of the bikebot and manipulator to enhance the stationary balance and posture control. The use of two-wheel steering and onboard manipulator enhances the balance capability. We first present a dynamic model of the system. A steering balance model is presented to analyze the steering configuration and maximize the balance capability. The balance condition is captured by an extended balance equilibrium manifold (BEM) of the mobile manipulation system. A BEM-enabled coordinated trajectory planning and control design is presented to achieve a balance-prioritized posture control. We conduct extensive experiments to validate and demonstrate the performance of the mechatronic and control design. The main contributions of this work are twofold. First, the presented two-wheel steering actuation analysis and model are innovative and provide a guidance on how to use the two-wheel steering design to increase balance capability of single-track mobile robots. It further explains the steering-induced balance capability differences between the single-track robot, such as bicycles and other two-wheel Segway-like balance robots. Second, the proposed coordinated motion control integrates the dynamic balance requirements with the task priority-based planning of a kinematic redundant manipulator. The extended BEM provides a new control approach to integrate the dynamic and kinematic constraints for mobile manipulation.

The rest of this article is organized as follows. Section II presents the problem statement and the systems dynamics. In Section III, we analyze the two-wheel steering mechanism and discuss the balance torque model. Section IV presents the coordinated pose control of the mobile manipulation. Experimental results are presented in Section V. Finally, Section VI concludes this article.

II. Problem Statement and Systems Dynamics

A. System Configuration and Problem Statement

Fig. 1(a) shows the prototype of the bikebot manipulation. Fig. 1(b) illustrates the side-view schematic of the kinematic configuration and Fig. 1(c) for a front-view of the system. An n-link lightweight manipulator with end-effector E is mounted on the bikebot body frame at point S. The two wheel/ground contact points are denoted as C_1 and C_2 and the wheelbase is denoted as l. Two sets of coordinate frames are introduced: Inertial frame $\mathcal{I}$ and body frame $\mathcal{F}_i$ for the ith manipulator link, $i = 1, \ldots, n$. Frame $\mathcal{F}_i$ is constructed by following the DH parameter convention [30]. $\mathcal{F}_0$ and $\mathcal{F}_n$ are for the base (platform) and end-effector frames, respectively. The horizontal and vertical distances from the bikebot's mass center G to C_1 are denoted as $l/2$ and h_G, respectively. The front and rear steering mechanisms are symmetric with same caster angle ε.

The bikebot's steering and roll angles are denoted as ϕ and φ_b, respectively. For both front and rear steering angles, the positive direction is defined as the counterclockwise about the steering axis. We define $\boldsymbol{\Theta} = [\theta_1 \cdots \theta_n]^\mathsf{T}$ as the manipulator joint angles. The generalized coordinates of the system are denoted as $\boldsymbol{q} = [\varphi_b \; \boldsymbol{\Theta}^\mathsf{T}]^\mathsf{T} \in \mathcal{Q} \subset \mathbb{R}^{n+1}$, where $\mathcal{Q}$ is admissible set for $\boldsymbol{q}$. We denote the pose (i.e., position and orientation) of end-effector E in $\mathcal{I}$ as $\boldsymbol{\xi}_e \in \mathbb{R}^6$.

Problem Statement: Given a set of N_ξ desired poses $\{\boldsymbol{\xi}_e^k\}_{k=1}^{N_\xi}$, $N_\xi \in \mathbb{N}$, the goal is to design a planning and control method for the bikebot manipulation (i.e., steering and joint angles control) to let end-effector E go through and hold stationary for short time at each $\boldsymbol{\xi}_e^k$, $k = 1, \ldots, N_\xi$.

B. Systems Dynamics

We use DH parameters $(\theta_i, d_i, a_i, \alpha_i)$ for the ith link of the manipulator, $i = 1, \ldots, n$. The homogeneous transformation matrix from $\mathcal{F}_i$ to $\mathcal{F}_0$ is written as [31]

$$\mathcal{T}_i(\boldsymbol{q}) = \mathcal{A}_1^0 \mathcal{A}_2^1 \cdots \mathcal{A}_i^{i-1} \tag{1}$$

where $\mathcal{A}_i^{i-1}$ denotes the transformation from $\mathcal{F}_i$ to $\mathcal{F}_{i-1}$ as

$$\mathcal{A}_i^{i-1} = \begin{bmatrix} \boldsymbol{R}_i^{i-1} & \boldsymbol{p}_i \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}. \quad (2)$$

$\boldsymbol{R}_i^{i-1} = \mathbf{R}_z(\theta_i)\mathbf{R}_x(\alpha_i)$, $\mathbf{R}_j(\beta) \in \mathrm{SO}(3)$, $j = x, y, z$, denotes the rotational matrix about j-axis with angle β, $\boldsymbol{p}_i = [a_i \cos\theta_i \ a_i \sin\theta_i \ d_i]^\mathsf{T}$ is the corresponding position vector in $\mathcal{F}_{i-1}$. With (1), we write the pose of end-effector E in $\mathcal{F}_0$ as $\boldsymbol{\xi}_e^{\mathcal{F}_0} = \boldsymbol{\xi}_e^{\mathcal{F}_0}(\mathcal{T}_n(\boldsymbol{q}))$.

We denote the mass center position of the ith link in $\mathcal{F}_0$ as $\boldsymbol{p}_{i_c}^0$ and its position in $\mathcal{I}$ is $\boldsymbol{p}_{i_c} = \boldsymbol{R}_0^{\mathcal{I}}(\boldsymbol{p}_0 + \boldsymbol{p}_{i_c}^0)$, where $\boldsymbol{p}_0$ is the position vector of point S in $\mathcal{I}$. The linear velocity $\boldsymbol{v}_{i_c}^0$ and angular velocity $\boldsymbol{\omega}_{i_c}^0$ of the ith link in $\mathcal{F}_0$ are obtained as

$$\dot{\boldsymbol{\xi}}_{i_c}^{\mathcal{F}_0} = \left[(\boldsymbol{v}_{i_c}^0)^\mathsf{T} \ (\boldsymbol{\omega}_{i_c}^0)^\mathsf{T} \right]^\mathsf{T} = \boldsymbol{J}_{i_c}\dot{\boldsymbol{\Theta}} \quad (3)$$

where $\boldsymbol{J}_{i_c} \in \mathbb{R}^{6\times n}$ is the Jacobian from $\mathcal{F}_i$ and $\mathcal{F}_0$. Therefore, the linear velocity $\boldsymbol{v}_{i_c}$ and angular velocity $\boldsymbol{\omega}_{i_c}$ in $\mathcal{I}$ are

$$\boldsymbol{v}_{i_c} = \boldsymbol{\omega}_b \times \boldsymbol{R}_0^{\mathcal{I}}(\boldsymbol{p}_0 + \boldsymbol{p}_{i_c}^0) + \boldsymbol{R}_0^{\mathcal{I}}\boldsymbol{v}_{i_c}^0, \boldsymbol{\omega}_{i_c} = \boldsymbol{\omega}_b + \boldsymbol{R}_0^{\mathcal{I}}\boldsymbol{\omega}_{i_c}^0 \quad (4)$$

where $\boldsymbol{\omega}_b = [\dot{\varphi}_b \ 0 \ 0]^\mathsf{T}$ is the platform roll angular velocity in $\mathcal{F}_0$.

The dynamic model of the mobile manipulation system is obtained through Lagrange's equations. The system's kinetic and potential energies are

$$T = T_b + \sum_{i=1}^{n} T_i, \ U = U_b + \sum_{i=1}^{n} U_i \quad (5)$$

where $T_b = \frac{1}{2}\boldsymbol{\omega}_b^\mathsf{T}\boldsymbol{I}_b\boldsymbol{\omega}_b + \frac{1}{2}m_b\boldsymbol{v}_G^\mathsf{T}\boldsymbol{v}_G$ is the kinetic energy for the bikebot and for the ith link of the manipulator $T_i = \frac{1}{2}m_i\boldsymbol{v}_{i_c}^\mathsf{T}\boldsymbol{v}_{i_c} + \frac{1}{2}\boldsymbol{\omega}_{i_c}^\mathsf{T}\boldsymbol{R}_i^{\mathcal{I}}\boldsymbol{I}_i(\boldsymbol{R}_i^{\mathcal{I}})^\mathsf{T}\boldsymbol{\omega}_{i_c}$; m_b and m_i are, respectively, the masses for the bikebot and the ith link, $\boldsymbol{v}_G$ is the velocity of the mass center G; $\boldsymbol{I}_b \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{I}_i \in \mathbb{R}^{3\times 3}$ are the inertia matrices for the bikebot about G and the ith link about its mass center, respectively. For potential energy terms in (5), for the bikebot, $U_b = m_b g(\boldsymbol{p}_G \cdot \boldsymbol{e}_z + \Delta h_G)$ and for the ith link, $U_i = m_i g \boldsymbol{p}_{i_c} \cdot \boldsymbol{e}_z$, where $\boldsymbol{p}_G$ is the position vector of G in $\mathcal{I}$, $g = 9.8$ m/s^2 is the gravitational constant, unit vector $\boldsymbol{e}_z = [0 \ 0 \ 1]^\mathsf{T}$, and Δh_G is the height change of G due to steering actuation [32].

The dynamic model is obtained by the Lagrangian method as

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau}$$

where $\boldsymbol{D}(\boldsymbol{q}) \in \mathbb{R}^{(n+1)\times(n+1)}$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{(n+1)\times(n+1)}$, and $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^{n+1}$ are the inertia, Coriolis, and gravitational matrices, respectively. We omit the details for these lengthy matrices. The generalized force $\boldsymbol{\tau} = [\tau_b \ \boldsymbol{\tau}_\theta^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{n+1}$ includes the controlled steering-induced balance torque τ_b and joint torque vector $\boldsymbol{\tau}_\theta \in \mathbb{R}^n$ for the manipulator. We further write the above model in the following block matrix form:

$$\begin{bmatrix} D_{bb} & \boldsymbol{D}_{b\theta} \\ \boldsymbol{D}_{\theta b} & \boldsymbol{D}_{\theta\theta} \end{bmatrix} \begin{bmatrix} \ddot{\varphi}_b \\ \ddot{\boldsymbol{\Theta}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_b \\ \boldsymbol{C}_\theta \end{bmatrix} \dot{\boldsymbol{q}} + \begin{bmatrix} G_b \\ \boldsymbol{G}_\theta \end{bmatrix} = \begin{bmatrix} \tau_b \\ \boldsymbol{\tau}_\theta \end{bmatrix} \quad (6)$$

where the block matrices are in appropriate dimensions and their dependencies on $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ are dropped for presentation brevity.

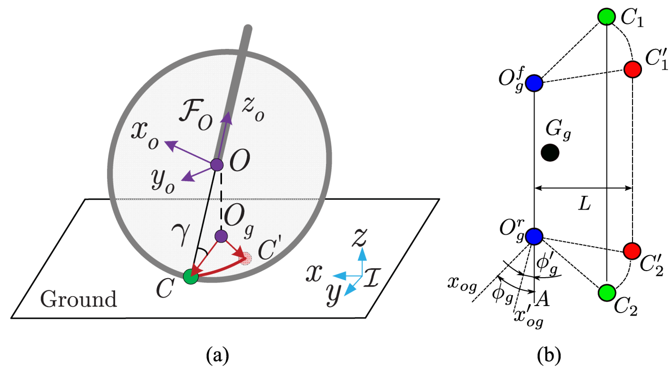


Fig. 2. Illustration of the steering mechanism and analysis. (a) Wheel plane geometry under steering angle increment δ. Wheel contact point changes from C to C'. (b) Wheel contact points C_1C_2 move to C_1' C_2' under a small steering angle increment δ and the geometric relationships between G_g and wheelbase line $C_1'C_2'$.

We will derive the steering-induced torque model for τ_b in the next section.

III. Steering Balance Model

In this section, we analyze the steering mechanism and derive a model to obtain the configuration that produces maximum steering-induced balance torque. Fig. 2(a) illustrates the schematic of the steering effect. We denote the wheel frame as $\mathcal{F}_O$ with wheel center O and the z_O-axis is along the steering axis and the y_O-axis is perpendicular to the wheel plane. The projection of O on the ground is denoted as O_g and the wheel/ground contact point as C. We consider quasi-static steering motion such that steering angle ϕ is built on an initial steering angle ϕ_0 with a small increment δ, namely, $\phi = \phi_0 + \delta$. Increment δ is small and the position change of point G under δ is negligible.

The orientation of the wheel plane with respect to frame $\mathcal{I}$ is approximately obtained by three successive rotations: First $-\phi$ about the z_O-axis, then $-\varphi_b$ about the x_O-axis, and finally $-\varepsilon$ about the y_O-axis. With this observation, we obtain the rotational transformation from $\mathcal{F}_O$ to $\mathcal{I}$ as

$$\boldsymbol{R}_{\mathcal{F}_O}^{\mathcal{I}} = \mathbf{R}_y(-\varepsilon)\mathbf{R}_x(-\varphi_b)\mathbf{R}_z(-\phi). \quad (7)$$

We denote the angle between the wheel plane and the ground as γ and it is straightforward to obtain

$$\cos\gamma = \boldsymbol{R}_{\mathcal{F}_O}^{\mathcal{I}}\boldsymbol{e}_y \cdot \boldsymbol{e}_z = \sin\phi\sin\varepsilon - \cos\phi\cos\varepsilon\sin\varphi_b \quad (8)$$

where unit vector $\boldsymbol{e}_y = [0 \ 1 \ 0]^\mathsf{T}$. To simplify the two-wheel steering design, both the front and rear steering angles, denoted, respectively, by ϕ^f and ϕ^r, are controlled and kept at symmetric position (i.e., same amplitude but opposite directions) for all time, namely, $\phi^f = -\phi^r = \phi$, with $\phi^f = \phi_{f0} + \delta$, $\phi^r = \phi_{r0} - \delta$, and $\phi_{f0} = -\phi_{r0}$. For brevity, we only use ϕ and δ in the following discussion. Fig. 2(b) illustrates the wheel/ground contact points. Under small δ, wheel/ground contact points C_1 and C_2 move to C_1' and C_2', respectively. With the above configuration, points C_1 and C_1' (C_2 and C_2') are located on a circular arc that is centered around O_g^f (O_g^r), projected points O^f (O^r) on the ground. The radii of the circular arc $\widehat{C_1\,C_1'}$ ($\widehat{C_2\,C_2'}$)

and the bikebot wheel are denoted as r and R, respectively. From the geometric relationship and (8), we obtain

$$r = R\cos\gamma = R\sin\phi\sin\varepsilon - R\cos\phi\cos\varepsilon\sin\varphi_b. \quad (9)$$

As shown in Fig. 2(b), under the same front and rear steering angles, wheelbase lines C_1C_2 and $C_1'C_2'$ are parallel. The corresponding projected steering angles are $\phi_g = \angle AO_g^r x_{og}$ and $\phi_g' = \angle AO_g^r x_{og}'$ (for the rear wheel). Let L denote the distance from the C_1C_2 to $O_g^f O_g^r$ and L is obtained by the geometry relationship as

$$L = r\cos\phi_g. \quad (10)$$

The relationship between ϕ_g and ϕ is captured by [32]

$$\phi_g = \arctan\left(\frac{\cos\varepsilon}{\cos\varphi_b}\tan\phi\right). \quad (11)$$

Given a fixed ϕ_0, from (9), $r = r_{\phi_0}$ is considered a constant value for a small roll angle (e.g., $\varphi_b \approx 0$) and, therefore, plugging (9) and (11) into (10), we obtain

$$L = r_{\phi_0}\frac{\cos\varphi_b}{\sqrt{\cos^2\varphi_b + \cos^2\varepsilon\tan^2\phi}} = L(\varphi_b, \phi_0, \delta)$$

where L is considered as a function of φ_b, ϕ_0, and δ.

We approximate the gravity-induced balance torque by using the above calculated L as the distance between G_g (the projected point of G on the ground) to $C_1'\ C_2'$ and obtain

$$\tau_b \approx m_b g L = mgL(\varphi_b, \phi_0, \delta). \quad (12)$$

It is helpful to find ϕ_0^* at which the increment δ generates the largest torque increase of τ_b. Thus, we introduce the steering torque sensitivity at $\varphi_b = 0$ as

$$S_\tau(\phi_0) = \left|\frac{\partial\tau_b}{\partial\delta}\right|_{\substack{\delta=0\\ \varphi_b=0}} = m_b g r_{\phi_0}\frac{\cos^2\varepsilon\tan\phi_0(\tan^2\phi_0+1)}{\left(\cos^2\varepsilon\tan^2\phi_0+1\right)^{3/2}}.$$

From the above equation, it is clear that at $\phi_0 = 0$, $S_\tau(\phi_0) = 0$ and this implies that the commonly used zero steering angle has the minimum steering torque sensitivity.

We further calculate that at $\phi_0^* = \frac{\pi}{2}$, $S_\tau(\phi_0)$ reaches its maximum value as $S_\tau(\phi_0^*) = m_b g R\tan\varepsilon$. Therefore, we focus on using $\phi_0^* = \frac{\pi}{2}$ for mobile manipulation control since it generates the largest balance torque per unit of steering angle. In this case, $\phi = \frac{\pi}{2} + \delta$ and $\phi_g = \frac{\pi}{2} + \delta_g$, the steering torque is calculated as

$$\tau_{b90} = m_b g r_{90}\cos\phi_g = -m_b g R\sin\varepsilon\cos\delta\sin\left(\tfrac{\delta}{\cos\varepsilon}\right) \quad (13)$$

where $r_{90} = R\sin\varepsilon\cos\delta$ is from (9) with $\varphi_b \approx 0$ and $\delta_g\cos\varepsilon \approx \delta$ is taken from (11). It is clear that a large caster angle configuration helps increase the steering-induced balance torque and, therefore, improve balance capability. For any other initial steering angle ϕ_0, the radius is calculated by (9) and the steering torque τ_b is obtained by (12).

It is interesting to note that under $\phi_0 = \frac{\pi}{2}$, the steering configuration is different with commonly used zero steering angle $\phi_0 = 0$. Indeed, the configuration is similar to double-track balance robot, such as Segway. This observation implies that double-track steering configuration, such as Segway-like robots helps provide more steering-induced balance torques than the single-track configuration, such as bicycles. We, therefore, use $\phi_0 = \frac{\pi}{2}$ in implementation.

IV. Coordinated Balance Control Design

A. Balance Equilibrium Manifold

The kinematics redundancy of the multi-DOFs manipulator enables the end-effector to reach the target poses with the balanced bikebot platform. If we consider the manipulator moves quasi-statically (i.e., slowly), the balanced bikebot roll angle φ_b and manipulator joint angles $\boldsymbol{\Theta}$ should satisfy an intrinsic relationship that is captured by BEM. From (6), the equation of motion of the bikebot is written as

$$D_{bb}\ddot{\varphi}_b + \boldsymbol{D}_{b\theta}\ddot{\boldsymbol{\Theta}} + \boldsymbol{C}_b\dot{\boldsymbol{q}} + G_b(\boldsymbol{q}) = \tau_b \quad (14)$$

where $G_b(\boldsymbol{q})$ is the total gravitational torque from the bikebot and the manipulator. Considering the quasi-static motion, namely, $\ddot{\boldsymbol{q}} = \dot{\boldsymbol{q}} = 0$, we define the BEM as

$$\mathcal{E} = \{\boldsymbol{q}_e = [\varphi_b^e\ \boldsymbol{\Theta}_e^\mathsf{T}]^\mathsf{T} : G_b(\boldsymbol{q}_e) = \tau_b,\ \boldsymbol{q} \in \mathcal{Q}\}. \quad (15)$$

The BEM captures all configurations that satisfy the static equilibrium constraint. Using BEM, we estimate the static maximum roll angle $\varphi_b^{\max}$ under the maximum balance steering $\tau_b^{\max}$.

To move the end-effector from one pose to another, a trajectory should be designed around the BEM at any time, namely, $\boldsymbol{q} \in \mathcal{E}$. A velocity constraint should be enforced given the BEM and limited steering actuation. Using (13), the steering torque is $\tau_b = -MgR\sin\varepsilon\cos\delta\sin\delta_g$, where $M = m_b + \sum_{i=1}^n m_i$ is the total mass of the entire system. Taking derivative of BEM condition $G_b(\boldsymbol{q}) = \tau_b$, we obtain

$$\dot{G}_b = \frac{\partial G_b}{\partial\boldsymbol{q}}\dot{\boldsymbol{q}} = -MgR\sin\varepsilon\frac{d}{d\delta}(\cos\delta\sin\delta_g)\dot{\delta} = h(\delta)\dot{\delta}$$

where $h(\delta) = -MgR\sin\varepsilon\frac{d}{d\delta}(\cos\delta\sin\delta_g)$. Defining $\boldsymbol{J}_G = \frac{\partial G_b}{\partial\boldsymbol{q}}$ as a Jacobian-like matrix, the above velocity constraint is specified as

$$|\boldsymbol{J}_G\dot{\boldsymbol{q}}| \le h_{\max}\dot{\delta}_{\max} \quad (16)$$

where $h_{\max} = \sup_\delta |h(\delta)|$ and $\dot{\delta}_{\max}$ is the maximum steering angular rate. Constraint (16) implies that when designing the trajectory $\boldsymbol{q}(t)$, the allowed motion velocity is restricted by the steering angular rate.

B. Balance-Prioritized Pose Trajectory Planning

The end-effector pose workspace in $\mathcal{I}$ is defined as

$$\mathcal{X}(\boldsymbol{q}) = \{\boldsymbol{\xi}_e : \boldsymbol{\xi}_e = \boldsymbol{\xi}_e(\mathcal{T}_{n+1}(\boldsymbol{q})), \boldsymbol{q} \in \mathcal{E}, |\tau_b| \le \tau_b^{\max}\} \quad (17)$$

where $\mathcal{T}_{n+1}(\boldsymbol{q})$ is the homogeneous transformation from $\mathcal{F}_n$ to $\mathcal{I}$. We further define the local end-effector pose workspace $\mathcal{X}_{\varphi_b^0}(\boldsymbol{\Theta}) \subseteq \mathcal{X}(\boldsymbol{q})$ under roll angle φ_b^0, $\boldsymbol{q}^0 = [\varphi_b^0\ \boldsymbol{\Theta}^\mathsf{T}]^\mathsf{T}$

$$\mathcal{X}_{\varphi_b^0}(\boldsymbol{\Theta}) = \left\{\boldsymbol{\xi}_e : \boldsymbol{\xi}_e = \boldsymbol{\xi}_e(\mathcal{T}_{n+1}(\boldsymbol{q}^0)), \boldsymbol{q}^0 \in \mathcal{E}\right\}.$$

The rationale to introduce $\mathcal{X}_{\varphi_b^0}(\boldsymbol{\Theta})$ is to specify the bikebot roll angle φ_b^0 first for balance task and then use the manipulator to achieve the pose control task. We consider the task priority from high to low as follows: 1) Bikebot platform balance; 2) pose control of end-effector E; and 3) collision avoidance during arm movement from one desired pose to another. Due to the redundant kinematics of the manipulator, we use a task-priority optimization approach to plan the trajectory.

We define the following balance-prioritized inverse kinematics (BPIK) problem. Given a sequence of desired end-effector poses $\boldsymbol{\xi}_e^k$, $k=1,\ldots,N_\xi$, the BPIK is to find optimal bikebot roll and manipulator joint angles $\boldsymbol{q}_k^*$ as

$$\boldsymbol{q}_k^* = \arg\min_{\boldsymbol{q}_k} \lambda_1\Gamma_1 + \lambda_2\Gamma_2 + \lambda_3\Gamma_3 \tag{18a}$$

$$\text{Subj. to } \lambda_4|G_b| \le \tau_b^{\max},\ \boldsymbol{q}\in\mathcal{E} \tag{18b}$$

where $\Gamma_1 = \|\boldsymbol{\xi}_e^k - \boldsymbol{\xi}_e(\mathcal{T}_{n+1}(\boldsymbol{q}_k))\|_2^2$, $\Gamma_2 = |G_b(\boldsymbol{q}_k) - G_b(\boldsymbol{q}_{k-1}^*)|^2$, $\Gamma_3 = \boldsymbol{e}_{k-1}^\mathsf{T}\boldsymbol{P}\boldsymbol{e}_{k-1}$, $\boldsymbol{e}_{k-1} = \boldsymbol{q}_k - \boldsymbol{q}_{k-1}^*$, $\lambda_1,\lambda_2,\lambda_3 > 0$, and $\lambda_4 > 1$ are weight parameters, $\boldsymbol{P} > 0$ is a symmetric positive definite matrix. We initialize with $\boldsymbol{q}_{-1}^* = \boldsymbol{0}$ and $G_b(\boldsymbol{q}_{-1}^*) = 0$.

Inequality (18b) is set as a hard constraint such that the steering output is always within the balance capability, while the pose regulation becomes a part of the objective function (i.e., Γ_1). Therefore, balance serves as a higher priority than pose regulation by the BPIK design. This is similar to the approach in [3], by projecting the low level priority task into subspace of solution of high level tasks. Terms Γ_2 and Γ_3 in (18a) try to minimize the difference between configurations at the current and the previous steps. We use the BPIK to obtain $\boldsymbol{q}_k^*$ from $\boldsymbol{\xi}_e^k$. We first search the solution in local workspace $\mathcal{X}_{\varphi_b^0}(\boldsymbol{\Theta})$ to avoid large bikebot movement. If this is impossible, the BPIK then searches the solution in the workspace $\mathcal{X}(\boldsymbol{q})$. If the calculated feasible poses are outside of $\mathcal{X}(\boldsymbol{q})$, (18) returns the closest results. Once obtaining the desired joint angles $\{\boldsymbol{q}_k^*\}_{k=1}^{N_\xi}$, we need to design transition trajectory along $\mathcal{E}$ between each two consecutive poses.

We consider a desired consecutive pair $(\boldsymbol{q}_{k-1}^*, \boldsymbol{q}_k^*)$ to position the end-effector. With user-specified starting and ending times denoted, respectively, as t_0 and t_f, we define $\boldsymbol{q}(t_0) = \boldsymbol{q}_{k-1}^*$ and $\boldsymbol{q}(t_f) = \boldsymbol{q}_k^*$. Motion trajectory $\boldsymbol{q}^*(t)$ needs to be designed from $\boldsymbol{q}(t_0)$ and $\boldsymbol{q}(t_f)$ along $\mathcal{E}$. The trajectory planning is formulated as the following optimization problem:

$$\min_{\boldsymbol{q}(t)} \int_{t_0}^{t_f} \boldsymbol{e}_{k-1}^\mathsf{T}\boldsymbol{W}_1\boldsymbol{e}_{k-1} + \dot{\boldsymbol{q}}^\mathsf{T}\boldsymbol{W}_2\dot{\boldsymbol{q}} + (\delta G_{b,k})^2 dt \tag{19a}$$

$$\text{Subj. to } \dot{\boldsymbol{q}}(t_0) = \dot{\boldsymbol{q}}_{k-1}^*, \dot{\boldsymbol{q}}(t_f) = \dot{\boldsymbol{q}}_k^*$$

$$\ddot{\boldsymbol{q}}(t_0) = \ddot{\boldsymbol{q}}_{k-1}^*, \ddot{\boldsymbol{q}}(t_f) = \ddot{\boldsymbol{q}}_k^* \tag{19b}$$

$$\boldsymbol{D}_{\theta b}\ddot{\varphi}_b + \boldsymbol{D}_{\theta\theta}\ddot{\boldsymbol{\Theta}} + \boldsymbol{C}_\theta\dot{\boldsymbol{q}} + \boldsymbol{G}_\theta = \boldsymbol{\tau}_\theta \tag{19c}$$

$$|\boldsymbol{J}_G\dot{\boldsymbol{q}}| \le h_{\max}\dot{\delta}_{\max}, \lambda_4|G_b| \le \tau_b^{\max} \tag{19d}$$

$$|\tau_{\theta,i}| \le \tau_{\theta,i}^{\max}, \boldsymbol{q}\in\mathcal{Q}, \|\dot{\boldsymbol{q}}\| \le \dot{q}_{\max}, \|\ddot{\boldsymbol{q}}\| \le \ddot{q}_{\max} \tag{19e}$$

where $\delta G_{b,k} = G_b(\boldsymbol{q}) - G_b(\boldsymbol{q}_{k-1}^*)$, $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{n+1}$ are positive diagonal matrices, and $\tau_{\theta,i}^{\max}$ is the maximum joint torque of the ith link, $i=1,\ldots,n$. To consider the quasi-static motion, the angular velocity and acceleration of the manipulator are bounded as in (19e). The constraint in (19d) is similar to that in (18) along with joint torque limits.

To solve (19), we try to avoid integration of the differential constraint (19c) and Bézier polynomials are used to specify the

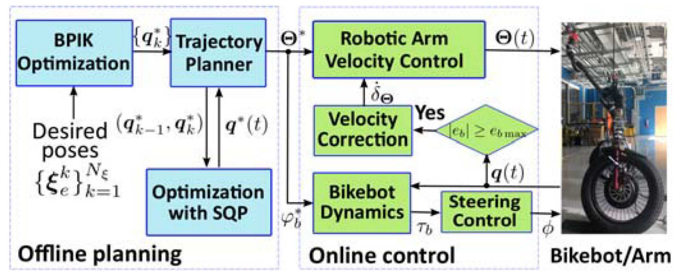


Fig. 3. Block diagram of the balance-prioritized trajectory planning and control scheme.

solution in each dimension of $\boldsymbol{q}(t)$. We use Bézier polynomial because of its attractive properties [33]. The solution $\boldsymbol{q}(t)$ is written in term of Nth-order Bézier polynomials ($N\in\mathbb{N}$) as

$$\varphi_b = b(s,\boldsymbol{p}_b) = \sum_{j=0}^N p_{b_j} b_j(s),\ \theta_i = b(s,\boldsymbol{p}_{\theta_i}) = \sum_{j=0}^N p_{\theta_{ij}} b_j(s)$$

for $i=1,\ldots,n$, where $b_j(s) = \frac{N!}{(N-j)!j!}(1-s)^{N-j}s^j$, parameters $\boldsymbol{p}_b = [p_{b_0}\cdots p_{b_N}]^\mathsf{T}$ and $\boldsymbol{p}_{\theta_i} = [p_{\theta_{i0}}\cdots p_{\theta_{iN}}]^\mathsf{T}$. The normalized progress variable $s = \frac{t-t_0}{t_f-t_0}$ maps $t\in[t_0,t_f]$ to $s\in[0,1]$.

From above formulation, we obtain $\boldsymbol{q}(t_0) = [b(0,\boldsymbol{p}_b)\, b(0,\boldsymbol{p}_{\theta_0})\,\cdots\, b(0,\boldsymbol{p}_{\theta_n})]^\mathsf{T}$, $\boldsymbol{q}(t_f) = [b(1,\boldsymbol{p}_b)\, b(1,\boldsymbol{p}_{\theta_0})\,\cdots\, b(1,\boldsymbol{p}_{\theta_n})]^\mathsf{T}$ and $\boldsymbol{q}(t)$ is then written as polynomials of s with parameters $\boldsymbol{p} = \{\boldsymbol{p}_b,\boldsymbol{p}_{\theta_1},\ldots,\boldsymbol{p}_{\theta_n}\}$. For $\dot{\boldsymbol{q}}(t)$, we obtain $\dot{\varphi}_b = \sum_{j=0}^N p_{b_j} b_j'(s)\frac{ds}{dt}$ and $\dot{\theta}_i = \sum_{j=0}^N p_{\theta_{ij}} b_j'(s)\frac{ds}{dt}$. Noting that $\frac{ds}{dt} = \frac{1}{t_f-t_0}$ is constant, $\dot{\boldsymbol{q}}(t)$ and $\ddot{\boldsymbol{q}}(t)$ are written in terms of $\boldsymbol{p}$. Therefore, the trajectory planning problem (19) is transformed into the s-domain and the differential constraints are written as algebraic formulation in polynomials of s and $\boldsymbol{p}$.

We discretize $s\in[0,1]$ with N_s sampling points and both the objective and constraint functions in (19) are evaluated at these points. A sequential quadratic programming (SQP) algorithm is then used to obtain the optimized trajectory $\boldsymbol{q}^*(t)$ [34]. In each iteration, a total of $3N_s(n+1)$ evaluations of $\boldsymbol{b}(s;\boldsymbol{p})$ (for $\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}$) and $N_s(n+2)$ evaluations of (19c) and (19d) are needed for $(n+1)(N-1)$ optimization variables. Additionally, the SQP solver has complexity $O(N^2n^2)$. Therefore, the computational complexity for solving (19) by the proposed approach is $O((N+N_s)Nn^2)$. As a comparison, a dynamic programming (DP) method can be used to solve (19) with complexity $O(N_s^2n^2)$. Because of $N_s \gg N$, the proposed method is much faster than the DP method. Algorithm 1 summarizes the trajectory planning, as described above.

C. Bikebot Steering and Manipulator Control

Fig. 3 illustrates the balance-prioritized trajectory planning and control design. The previous section discusses the trajectory planner to obtain $\boldsymbol{q}^*(t)$ for a given set $\{\boldsymbol{\xi}_e^k\}_{k=1}^{N_\xi}$. We present the controller design to follow $\boldsymbol{q}^*(t)$.

We first present the steering control of the bikebot to follow φ_b^* and then the manipulator controller to follow $\boldsymbol{\Theta}^*$. Using (14), we define the roll angle error $e_b = \varphi_b - \varphi_b^*$ and a feedback

Algorithm 1: Trajectory Planning for Pose Regulation.

Specify $\{\boldsymbol{\xi}_e^k\}_{k=1}^{N_\xi}$, $\lambda_i, i = 1, ..., 4$, $\boldsymbol{P}$, $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, and $\epsilon > 0$;
for $k \le N_\xi$ **do**
  if $k = 1$ **then**
    $\boldsymbol{q}_{k-1}^* \leftarrow \boldsymbol{0}$, $G(\boldsymbol{q}_{k-1}^*) \leftarrow 0$;
    Solve $\boldsymbol{q}_k^* \in \mathcal{X}(\boldsymbol{q})$ by (18) with $\boldsymbol{\xi}_e^k$;
  else
    Solve $\boldsymbol{q}_k^* \in \mathcal{X}_{\varphi_{b,k-1}^*}(\boldsymbol{\Theta})$ by (18) with $\boldsymbol{\xi}_e^k$;
    Calculate the pose $\boldsymbol{\xi}_e(\boldsymbol{q}_k^*)$;
    if $\|\boldsymbol{\xi}_e^k - \boldsymbol{\xi}_e(\mathcal{T}_{n+1}(\boldsymbol{q}_k^*))\| \ge \epsilon$ **then**
      Re-calculate $\boldsymbol{q}_k^* \in \mathcal{X}(\boldsymbol{q})$ by (18);
  $k \leftarrow k + 1$
for $j \le N_\xi$ **do**
  Specify t_0 and t_f; $\boldsymbol{q}(t_0) \leftarrow \boldsymbol{q}_{j-1}^*$, $\boldsymbol{q}(t_f) \leftarrow \boldsymbol{q}_j^*$;
  Plan $\boldsymbol{q}^*(t)$, $t \in [t_0, t_f]$ by (19) using Bézier polynomial specification;
  $j \leftarrow j + 1$;

linearization control is designed as

$$\tau_b = D_{bb}\ddot{\varphi}_b^* + \boldsymbol{D}_{b\theta}\ddot{\boldsymbol{\Theta}} + \boldsymbol{C}_b\dot{\boldsymbol{q}} + G_b + k_p e_b + k_d \dot{e}_b \tag{20}$$

where $k_p, k_d > 0$ are the feedback gains. From (20), we use the steering balance model (12) to obtain the steering angle ϕ.

For the manipulator, we take the velocity control to follow the desired trajectory $\boldsymbol{\Theta}^*$ in $\mathcal{F}_0$. We recognize $\ddot{\boldsymbol{\Theta}} \approx 0$ after the system compensating for the gravitational term ($\boldsymbol{G_\theta}$). We extend the velocity control in [10] and [35] with additional velocity correction under bikebot roll motion errors. Defining joint angle error $\boldsymbol{e}_\Theta = \boldsymbol{\Theta} - \boldsymbol{\Theta}^*$, the velocity control in the joint workspace is given by

$$\dot{\boldsymbol{e}}_\Theta = -\boldsymbol{K}_p \boldsymbol{e}_\Theta + I_\Theta \dot{\boldsymbol{\delta}}_\Theta \tag{21}$$

where $\boldsymbol{K}_p = \operatorname{diag}\{K_{p1}, \ldots, K_{pn}\}$ with $K_{pi} > 0, i = 1, \ldots, n$ and $I_\Theta = 1$ if $|e_b| > \varepsilon_b$; otherwise $I_\Theta = 0$, with an error threshold $\varepsilon_b > 0$. The velocity correction $\dot{\boldsymbol{\delta}}_\Theta$ in (21) is designed as

$$\dot{\boldsymbol{\delta}}_\Theta = -\kappa \frac{\partial \delta G_b}{\partial \boldsymbol{\Theta}} = 2\kappa \left[G_b(\boldsymbol{q}^*) - G_b(\boldsymbol{q})\right] \left(\frac{\partial G_b(\boldsymbol{q})}{\partial \boldsymbol{\Theta}}\right)^{\mathsf{T}}$$

where $\delta G_b = (G_b(\boldsymbol{q}^*) - G_b(\boldsymbol{q}))^2$ denotes deviation from the BEM and $\kappa > 0$ is a scalar.

Under (20), the closed-loop roll error dynamics is

$$\ddot{e}_b + k_d \dot{e}_b + k_p e_b = 0 \tag{22}$$

and $e_b(t)$ converges to zero exponentially. Without loss of generality, let $k_d^2 < 4k_p$ and then from (22) we obtain

$$|e_b(t)| \le M_b e^{-\frac{k_d}{2}t} \tag{23}$$

where $M_b > 0$ is a finite constant that is related to $e_b(0)$ and $\dot{e}_b(0)$. Therefore, for any $t \ge t_b := \frac{2}{k_d}\ln(\frac{M_b}{\varepsilon_b})$, $|e_b(t)| \le \varepsilon_b$. To show the convergence for $\boldsymbol{e}_\Theta$, we consider a Lyapunov function candidate $V(t) = \boldsymbol{e}_\Theta^{\mathsf{T}}\boldsymbol{e}_\Theta = \|\boldsymbol{e}_\Theta\|^2 > 0$ for any nonzero error $\boldsymbol{e}_\Theta$. Letting $l_\Theta := \sup_{0 \le t \le t_b} \|I_\Theta \dot{\boldsymbol{\delta}}_\Theta\|$, we have

$$\dot{V}(t) = -2\boldsymbol{e}_\Theta^{\mathsf{T}}\boldsymbol{K}_p\boldsymbol{e}_\Theta + 2I_\Theta \boldsymbol{e}_\Theta^{\mathsf{T}}\dot{\boldsymbol{\delta}}_\Theta \le -2\lambda_p\|\boldsymbol{e}_\Theta\|^2 + 2l_\Theta\|\boldsymbol{e}_\Theta\|$$

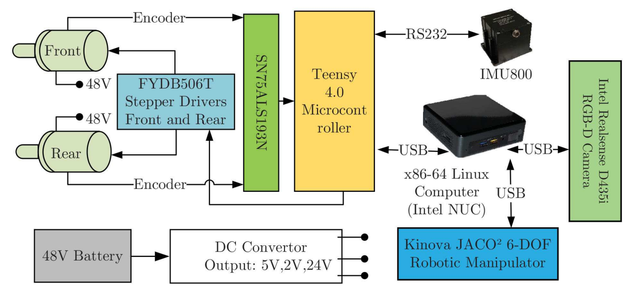


Fig. 4. Schematic of the interconnection among sensors, actuators and embedded systems for the bikebot-manipulator system.

where $\lambda_p = \min_{1 \le i \le n} K_{pi}$. We introduce $W(t) = \sqrt{V(t)} = \|\boldsymbol{e}_\Theta(t)\|$ and from the above inequality, we obtain $\dot{W} \le -\lambda_p W + l_\Theta$. Thus, we have $\frac{d}{dt}(We^{\lambda_p t}) = \dot{W}e^{\lambda_p t} + W\lambda_p e^{\lambda_p t} \le l_\Theta e^{\lambda_p t}$ and integrating from 0 to t, we obtain

$$W(t)e^{\lambda_p t} - W(0) \le \frac{l_\Theta}{\lambda_p}(e^{\lambda_p t} - 1).$$

Noting $W(t) = \|\boldsymbol{e}_\Theta(t)\|$, the above inequality becomes

$$\begin{aligned} \|\boldsymbol{e}_\Theta(t)\| &\le \|\boldsymbol{e}_\Theta(0)\|e^{-\lambda_p t} + \frac{l_\Theta}{\lambda_p}(1 - e^{-\lambda_p t}) \\ &\le \|\boldsymbol{e}_\Theta(0)\|e^{-\lambda_p t} + \frac{l_\Theta}{\lambda_p}. \end{aligned} \tag{24}$$

From the above analysis, $\dot{V}$ is negative outside of the compact set $\mathcal{S} = \{\boldsymbol{e}_\Theta : \|\boldsymbol{e}_\Theta(t)\| \le \frac{l_\Theta}{\lambda_p}\}$ and $\boldsymbol{e}_\Theta(t)$ exponentially converges to $\mathcal{S}$. Note that for $t \ge t_b$, $I_\Theta = 0$, $\boldsymbol{e}_\Theta(t)$ converges to zeros exponentially due to stable dynamics (21).

With error convergence in (23) and (24), we obtain the error bound for $\boldsymbol{e}_q = \boldsymbol{q} - \boldsymbol{q}^* = [e_b \; \boldsymbol{e}_\Theta^{\mathsf{T}}]^{\mathsf{T}}$ as

$$\begin{aligned} \|\boldsymbol{e}_q(t)\| &= \sqrt{e_b^2(t) + \|\boldsymbol{e}_\Theta(t)\|^2} \le |e_b(t)| + \|\boldsymbol{e}_\Theta(t)\| \\ &\le M_b e^{-\frac{k_d}{2}t} + \|\boldsymbol{e}_\Theta(0)\|e^{-\lambda_p t} + \frac{l_\Theta}{\lambda_p}. \end{aligned} \tag{25}$$

Considering $\boldsymbol{e}_\xi = \boldsymbol{\xi}_e(\boldsymbol{q}) - \boldsymbol{\xi}_e(\boldsymbol{q}^*)$ and $\boldsymbol{q} = \boldsymbol{q}^* + \boldsymbol{e}_q$, we have

$$\boldsymbol{e}_\xi = \boldsymbol{\xi}_e(\boldsymbol{q}^*) + \left.\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}^*}\boldsymbol{e}_q + \boldsymbol{\Delta}_q - \boldsymbol{\xi}_e(\boldsymbol{q}^*) = \boldsymbol{J}_e(\boldsymbol{q}^*)\boldsymbol{e}_q + \boldsymbol{\Delta}_q \tag{26}$$

where $\boldsymbol{J}_e(\boldsymbol{q}) \in \mathbb{R}^{6\times(n+1)}$ is the Jacobian matrix from $\mathcal{F}_n$ to $\mathcal{I}$ and $\boldsymbol{\Delta}_q \in \mathbb{R}^6$ is the higher order term of error $\boldsymbol{e}_q$. From (26), it is straightforward to obtain that $\|\boldsymbol{e}_\xi(t)\| \le \|\boldsymbol{J}_e(\boldsymbol{q}^*)\|\|\boldsymbol{e}_q(t)\| + \|\boldsymbol{\Delta}_q\|$. For the higher order term $\|\boldsymbol{\Delta}_q\| = O(\|\boldsymbol{e}_q\|^2)$, there exists a finite constant $M_\delta > 0$ such that $\|\boldsymbol{\Delta}_q\| \le M_\delta\|\boldsymbol{e}_q(t)\|$ due to (25) and then $\|\boldsymbol{e}_\xi(t)\| \le M_q\|\boldsymbol{e}_q(t)\|$, where $M_p = \sup_{\boldsymbol{q}^*}\|\boldsymbol{J}_e(\boldsymbol{q}^*)\| + M_\delta$. Therefore, the pose error $\boldsymbol{e}_\xi(t)$ converges to a small ball near zero exponentially and the robotic system is stable.

V. Experiments

A. Experiment Setup

Fig. 1(a) shows the prototype of the two-wheel-steered bikebot with an onboard six-DOF robotic manipulator (Jaco2 from Kinova Inc.). Fig. 4 illustrates the interconnection schematic of the embedded systems and actuators and sensors. Both the

TABLE I
VALUES FOR THE MODEL PARAMETERS OF THE BIKEBOT PLATFORM

m_b (kg)	I_b (kgm^2)	h_G (m)	l (m)	ε (deg)	R (m)
46.9	3.2	0.53	1.2	20	0.3

TABLE II
DH PARAMETER VALUES AND INERTIA PARAMETERS OF THE SIX-DOF ROBOTIC MANIPULATOR

Link	α_i (deg)	a_i (m)	d_i (m)	m_i (kg)	(I_{xx}, I_{yy}, I_{zz}) (kgm^2)
1	90	0	0.276	1	$(0.0022, 0.0006, 0.0023)$
2	180	0.41	0	1.5	$(0.0041, 0.0255, 0.0217)$
3	90	0	-0.01	0.8	$(0.0029, 0.0027, 0.0004)$
4	60	0	-0.25	0.3	$(0.7085, 0.7405, 0.1782)$
5	60	0	-0.009	0.3	$(0.8275, 0.8520, 0.1708)$
6	180	0	0.203	0.6	$(0.0048, 0.0048, 0.0002)$

Fig. 5. Experimental setup for validation of the steering mechanism and balance torque model.

front and the rear wheels can be steered around 360 degs by two stepper motors. A real-time low-level embedded system (Teensy 4.0 microcontroller) is used for the steering motor control, while the robotic manipulator is controlled by a powerful small-size computer (Intel NUC module) with robot operating system (ROS). One inertial measurement unit (IMU) (model 800 from Motion Sense Inc.) is mounted at the upper frame of the bikebot to measure the roll angle. The front and rear steering angles are measured by two encoders and the manipulator joint angles are obtained by the embedded encoders. The real-time bikebot steering control and data acquisition frequency was implemented at 100 Hz and the low-level manipulator velocity control was run at 1000 Hz.

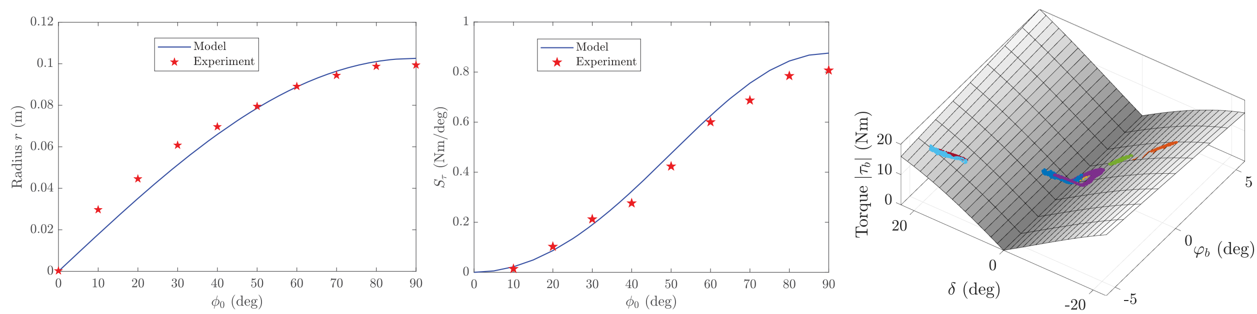
Table I lists the model parameter values for the bikebot, where I_b is the mass moment of inertia of the bikebot about the wheelbase. Table II lists the values of the DH parameters and mass moments of inertia of the manipulator links $\boldsymbol{I}_i = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz})$ about their mass centers. The other physical parameters for each link can be found in [36]. To validate the steering mechanism and models, we also built and conducted experiments to measure the tire/ground contacts and movement and Fig. 5 shows the experimental setup. A motion capture system (4 Vantage cameras from Vicon Ltd.) was used to measure the angle and contact points between the wheel plane and the ground at different angles ϕ_0.

B. Experimental Results

We first present the validation of the steering balance models. Fig. 6(a) and (b) shows the values of the turning radius r_{ϕ_0} and the steering torque sensitivity S_τ, respectively, as steering angle ϕ_0 increases. The experiment data clearly confirm the model predictions. It is clear that when $\phi_0 = 90$ degs, the projected radius r_{90} reaches the maximum value and the increasing trend is monotonic. The steering sensitivity S_τ also reaches its maximum point around $\phi_0 = 90$ degs with $S_\tau = 0.87$ Nm/deg. At $\phi_0 = 0$, the projected radius r_0 and steering torque sensitivity S_τ are near zero. From this observation, an initial steering angle ϕ_0 is chosen around 90 degs for the following experiments. At $\phi_0 = 90$ degs, multiple stationary balancing experiments were conducted. Fig. 6(c) shows the steering-induced balance torque τ_b at different roll angles φ_b and increments δ. Multiple experimental trials are plotted together with the steering torque model prediction from (13), i.e., the 3-D surface, as shown in the figure. The experimental data are scattered around the torque model prediction with small errors. These results validate the steering-induced balance torque model.

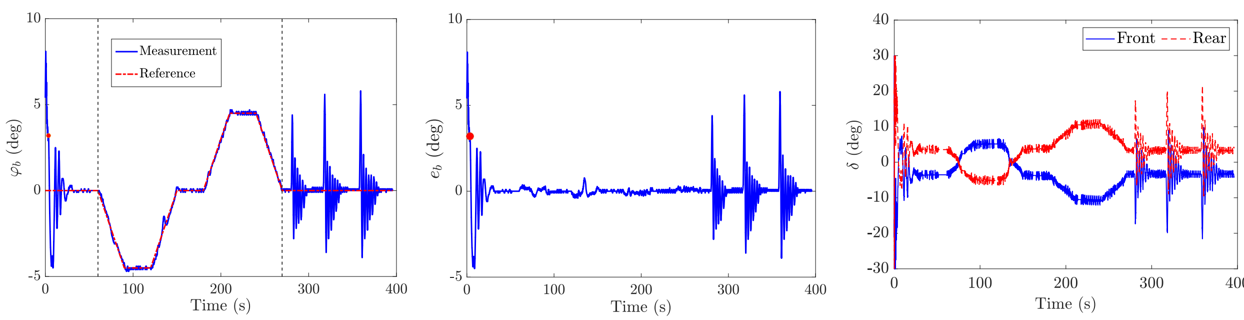
Fig. 7 shows the bikebot balance control results. The manipulator was removed from the bikebot in this experiment. The controller (20) was used with feedback gains $k_p = 8.5$ and $k_d = 2$. The entire trial is divided into three stages as separated by the vertical lines in Fig. 7(a). Fig. 7(b) shows the roll angle tracking errors. Fig. 7(c) illustrates the front and rear wheel steering angle increments. In the first stage, the initial roll angle was about 4 degs and it was then regulated around zero. A chattering phenomenon was observed in the roll angle profile and this was due to the fact that the IMU angular measurement resolution was around 0.1 deg, that is, the IMU measurement was discretized with a minimal resolution of 0.1 deg. This oscillation also caused a similar chattering behavior in steering angle increments in Fig. 7(c) since the roll angle measurement was used in steering control. In the second stage starting at around $t = 60$ s, the bikebot was commanded to move around zero with the maximum roll angles around 4.5 degs. The change of the reference roll angle was slow to meet the quasi-static movement. The tracking error approached to zero. In the third stage starting around $t = 270$ s, multiple disturbances were applied by manually pushing the upper frame of the bikebot. The roll angle errors caused by the disturbances reached 6 degs and the steering actuation compensated for the disturbances. These results demonstrate the steering balance control performance. Since the design enforced symmetrical steering commands, the front and rear steering angles responses showed highly similar behaviors.

Using the BEM and model parameters, we estimate the maximum stationary balance roll angles. With one-wheel steering control, the maximum balanced roll angle is around 3.4 degs; with two-wheel steering control, around 5.6 degs; and additionally, if the manipulator is used to help balance collaboratively, it increases to 11.6 degs. To validate these estimates, we conducted



(a) (b) (c)

Fig. 6. Experimental results for the steering torque model. (a) Comparison results of the model prediction with the experiments for radius r_{ϕ_0}. (b) Comparison of the steering sensitivity model prediction with the experiments. (c) Comparison of the balance torque model prediction with the experiments under steering angle increment δ and roll angle φ_b with $\phi_0 = 90$ degs.



(a) (b) (c)

Fig. 7. Bikebot balance control experimental results. (a) Bikebot roll angle φ_b. (b) Bikebot roll angle error e_b. (c) Steering angle increments. The red markers "•" in (a) and (b) indicate that the initial angles.

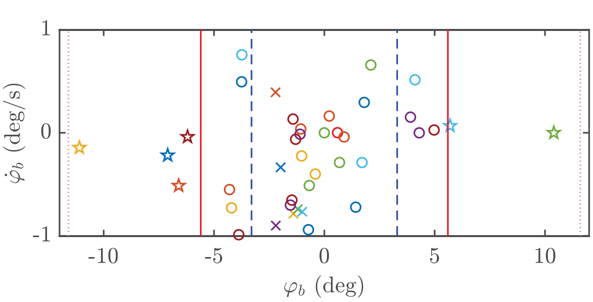
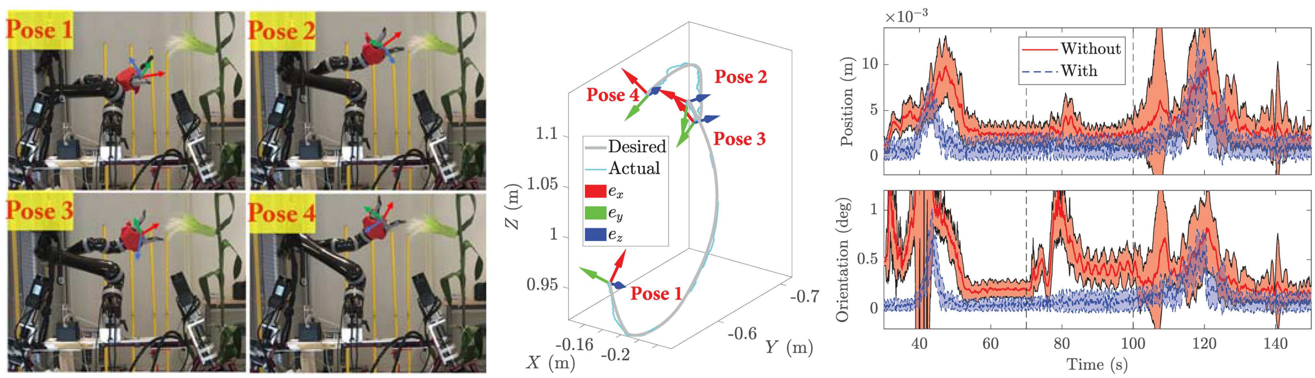


Fig. 8. Verification of the recoverable roll angle region. The markers "×," "○," and "⋆" represent the successful balance trial of the bikebot by one-wheel steering, two-wheel steering, and collaboratively two-wheel steering with manipulation balancing strategies, respectively. The vertical lines indicate the estimated maximum angle boundaries $\varphi_b = \pm 3.4$, ± 5.6, and ± 11.6 degs.

multiple balance control experiments. One experimental trial was considered successful if the system was kept balanced for a time duration over 50 s. Fig. 8 shows the successful trials in the $\dot{\varphi}_b$-φ_b plane. Each marker in the figure represents the state at which the bikebot successively started to balance under steering control, which was confirmed by comparing the model predictions from (13) and (14). The experiments, which are in agreement with the model prediction, validate the model analysis and demonstrate the balance capability under various bikebot balancing strategies.

We now present a plant inspection example for the end-effector to continuously go through and stop momentarily at four poses (i.e., $N_\xi = 4$). This represents the end-effector movement during a plant scanning and inspection task [19]. Fig. 9(a) shows the snapshots of the end-effector at the four poses. The major movement of the end-effector (with a mounted camera) was along the z-axis in $\mathcal{I}$ and orientation always pointed toward to the stalk of a fake corn plant. The end-effector moved from one pose to another in sequence and stopped for about 15 s at each pose. Fig. 9(b) shows the 3-D trajectories with the four poses. The planning and control parameters used in experiments include: $\lambda_1 = 10$, $\lambda_2 = 1$, $\lambda_3 = 5$, $\lambda_4 = 1.5$, $\boldsymbol{W}_1 = \mathrm{diag}(10, 5, 5, 5, 1, 1, 1)$, $\boldsymbol{W}_2 = \mathrm{diag}(1, 1, 1, 1, 1, 1)$, $\kappa = 5$, $\epsilon = 0.1$, $\varepsilon_b = 0.4$ degs, $\dot{q}_{\max} = 36$ deg/s, $\ddot{q}_{\max} = 120$ deg/s^2, $\boldsymbol{\tau}_\theta^{\max} = [10\ 15\ 10\ 5\ 5\ 5]^\mathsf{T}$ Nm, $\delta_{\max} = 15$ degs, $\dot{\delta}_{\max} = 20$ deg/s, and $N = 7$. For off-line planning implementation, the number of data points was chosen as $N_s = 50$ in each dimension of $\boldsymbol{q}(t)$ and the SQP method (via *fmincon* function) in Matlab was used for solving (19). The obtained Bézier polynomial trajectory was then sampled at 100 Hz for real time control. The off-line planner computed the trajectory for the next pose transition when conducting motion control of the current pose movement. By doing so, the proposed planner was capable to obtain the trajectory with fast computational time. Table III lists the desired end-effector poses, the poses planned by the BPIK and the actual poses. It is clear



(a) (b) (c)

Fig. 9. (a) Snapshots of the manipulator with an inspected plant at four poses. (b) Pose transition and end-effector trajectory under the pose planning and control. The arrows (i.e., unit vectors $(\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z)$) at each pose represent the actual orientation of the end-effector in $\mathcal{I}$. (c) Pose error (mean with one standard deviation) in transition process (from Poses 1 to 4 as indicated by the vertical lines) from 15 experiment trials. Top: Position error; Bottom: Orientation error. The video of the experiment can be found at https://youtu.be/jHQRNrrnPMc.

TABLE III
COLLABORATIVE END-EFFECTOR POSE CONTROL RESULT

Pose	Desired $\boldsymbol{\xi}_e^k$	BPIK-planned $\boldsymbol{\xi}_e(\mathcal{T}_{n+1}(\boldsymbol{q}_k^*))$	Actual $\boldsymbol{\xi}_e^k$	Errors	
				Position	Orientation
1	$[-13\ \ -55\ 94\ \ -61\ 52\ 96]^\mathsf{T}$	$[-14\ \ -56\ 95\ \ -64\ 52\ 98]^\mathsf{T}$	$[-14\ \ -56\ 95\ \ -63\ 51\ 97]^\mathsf{T}$	0.76 ± 0.02	0.43 ± 0.005
2	$[-15\ \ -67\ 109\ 57\ 31\ \ -57]^\mathsf{T}$	$[-15\ \ -65\ 106\ 58\ 30\ \ -58]^\mathsf{T}$	$[-15\ \ -67\ 109\ 53\ 30\ \ -58]^\mathsf{T}$	0.34 ± 0.01	0.16 ± 0.017
3	$[-17\ \ -69\ 106\ 38\ 55\ \ -37]^\mathsf{T}$	$[-16\ \ -68\ 107\ 37\ 55\ \ -37]^\mathsf{T}$	$[-16\ \ -68\ 107\ 37\ 55\ \ -37]^\mathsf{T}$	0.72 ± 0.13	0.40 ± 0.08
4	$[-15\ \ -58\ 114\ 32\ 48\ \ -21]^\mathsf{T}$	$[-15\ \ -61\ 112\ 39\ 42\ \ -18]^\mathsf{T}$	$[-15\ \ -60\ 112\ 39\ 42\ \ -18]^\mathsf{T}$	0.34 ± 0.09	0.15 ± 0.046

The unit for position is cm and for orientation is deg. The error mean and standard deviation values are calculated from 10 S data of the pose holding phase.

that the position errors are within 8 mm and orientation errors within 0.45 degs at these four poses.

Fig. 10 shows the detailed experimental results. Fig. 10(a) and (b) shows the bikebot roll angle φ_b and the six joint angles $\boldsymbol{\Theta}$ of the manipulator, respectively. Since φ_b and the first three joint angles (θ_1-θ_3) played a major role to balance the entire system, their reference trajectories were designed to avoid large variations. Poses 2 and 3 were searched in the local workspace $\mathcal{X}_{\varphi_{b1}^0}(\boldsymbol{\Theta})$, and Pose 4 is searched in the workspace $\mathcal{X}(\boldsymbol{q})$. The bikebot roll angle change was approximately around 1.5 degs. At $t = 0$ s, the manipulator was at the desired balance configuration as Pose 1. Around $t = 30$ s, the manipulator started moving to Pose 2. Small disturbances were introduced at around $t = 40$ s, causing about a 0.4-deg roll angle error. The velocity correction control was applied to compensate for the roll angle error; see Fig. 10(e). No obvious roll angle error was observed during the transition from Poses 2 to 3 (except around 0.1 degs oscillation). The bikebot platform was required to move in the transition from Poses 3 to 4. Around $t = 110$ s, a large roll angle change was commanded by the steering actuation and the velocity correction control was needed; see Fig. 10(e).

We further repeated the above four-pose control experiment 15 times. Fig. 9(c) summarizes the statistics (i.e., mean and one-standard deviation) of the end-effector pose errors during the motion. The end-effector position errors are less than 5 mm and the orientation errors within 0.3 degs. Relative large errors happened around $t = 42$ and 120 s with about 10 mm and 0.7 degs, respectively. This is consistent with the previous results.

TABLE IV
COMPUTATIONAL COST COMPARISON FOR THE PROPOSED AND THE DP ALGORITHMS

N_s	50	100	200	500
Proposed algorithm (s)	9.69	14.94	31.58	21.46
DP method (hour)	0.14	1.04	3.02	≥ 12

The position errors are at the same level of the manipulator hardware performance limits (3.7 mm) that are provided by the vendor and the orientation errors are much less that level (2.1 degs). The results demonstrate the successful balance and pose control performance by the design.

C. Discussion

To further demonstrate the performance, we conducted additional comparison experiments. Fig. 10(a) also includes the bikebot roll angle when the balance priority was not enforced in trajectory planning. Clearly, the entire system lost balance in the pose transition phase at $t = 50$ s. This confirms the effectiveness of the priority-based task control. In Fig. 9(c), we also present the errors statistics (i.e., mean and one-standard deviation) without the online velocity correction. In this case, both the position and orientation tracking errors are larger than these results under the velocity control correction. We also conducted computational time comparison between the proposed trajectory planning algorithm and the DP method. Table IV shows the comparison results. The numerical results confirmed that the computational

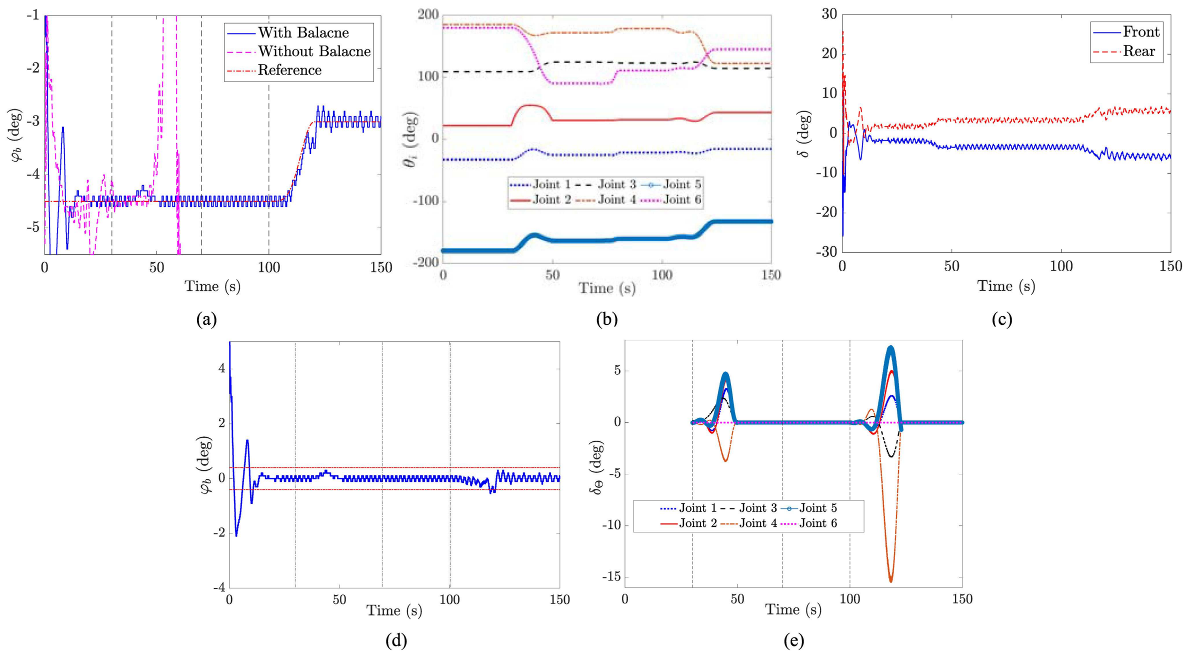


Fig. 10. Large roll angle balance control of the mobile manipulation system. (a) Roll angle, (b) robotic arm joint angles, and (c) steering angle increments. The vertical lines in (a) divides the entire process into four phases for each pose. (d) Roll angle error. (e) Online robotic arm trajectory correction in pose transition phase (the difference between off-line planing results and the actual angle).

cost of the DP method was over 200 times higher than that of the proposed algorithm to solve the optimization problem in (19).

Although demonstrating successful results, the current work have several limitations for further improvement. We only studied coordinated control of the manipulator on stationary bikebot and it would be desirable to extend to moving platform case. Second, the steering control does not include the dynamic effects of the steering mechanism. The control performance might be improved with incorporating dynamic steering effect. The trajectory of the bikebot roll angle and the manipulator joint angles was planned off-line and online planning is desirable for applications with dynamic obstacle avoidance. Finally, the proposed method is built on the precise robot model and it is desirable to extend to handle model uncertainties in complex, dynamic environment. One possible approach is to use machine learning-based methods. For example, as discussed in [37], the robot dynamics might be approximated and estimated using a Gaussian process model and a learning-based motion control can be then designed.

VI. Conclusion

In this article we presented a coordinated balance and pose control for a stationary mobile manipulation using a two-wheel steered bikebot. The mobile platform is inherently unstable and the dynamics of the platform and the manipulator are strongly coupled. We presented a two-wheel steering model and identified the use of $\phi_0 = 90$ degs as the most beneficial steering angle for stationary balance. A balance equilibrium manifold was extended to the mobile manipulation for coordinated motion control. Built on the BEM, a balance-priority design was then presented to solve the optimal joint angles for the bikebot and the manipulator. Coordinated balance and pose control was achieved by enforcing the entire system moving on the BEM with online manipulator velocity correction control. We conducted experiments and the results demonstrated the performance of the balance and pose control for a plant inspection application.

References

[1] Z. Lu, A. Chauhan, F. Silva, and L. S. Lopes, "A brief survey of commercial robotic arms for research on manipulation," in *Proc. IEEE Symp. Robot. Appl.*, 2012, pp. 986–991.

[2] O. Brock, J. Park, and M. Toussaint, "Mobility and Manipulation," in *Springer Handbook of Robotics*. B. Siciliano and O. Khatib, Eds. Berlin, Germany: Springer, 2016.

[3] E. Simetti, G. Casalino, F. Wanderlingh, and M. Aicardi, "A task priority approach to cooperative mobile manipulation: Theory and experiments," *Robot. Auton. Syst.*, vol. 122, 2019, Art. no. 103287.

[4] B. Arad *et al.*, "Development of a sweet pepper harvesting robot," *J. Field Robot.*, vol. 37, no. 6, pp. 1027–1039, 2020.

[5] K.-D. Nguyen, "Stability and robustness of adaptive controllers for underactuated Lagrangian systems and robotic networks," Ph.D. dissertation, Dept. Mech. Sci. Eng., Univ. Illinois, Urbana-Champaign, IL, USA, 2015.

[6] H. Stuart, S. Wang, O. Khatib, and M. R. Cutkosky, "The ocean one hands: An adaptive design for robust marine manipulation," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 150–166, 2017.

[7] S. Kim, H. Seo, S. Choi, and H. J. Kim, "Vision-guided aerial manipulation using a multirotor with a robotic arm," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 4, pp. 1912–1923, Aug. 2016.

[8] H. Bonyan Khamseh, F. Janabi-Sharifi, and A. Abdessameud, "Aerial manipulation—A literature survey," *Robot. Auton. Syst.*, vol. 107, pp. 221–235, 2018.

[9] P. Holmes *et al.*, "Reachable sets for safe, real-time manipulator trajectory design," in *Proc. Robot, Sci. Syst.*, Oregon, USA, 2020, pp. 1–9.

[10] R. Raja, A. Dutta, and B. Dasgupta, "Learning framework for inverse kinematics of a highly redundant mobile manipulator," *Robot. Auton. Syst.*, vol. 120, pp. 1–13, 2019.

[11] J. Woolfrey, "Control of manipulators on moving platforms under disturbance," *Ph.D. dissertation Dept. Eng. and Info. Tech., Univ. Tech.* Sydney, Sydney, Australia, 2019.

[12] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, “BigDog, the rough-terrain quadruped robot,” in *Proc. Int. Federation Accountants World Congr.*, Seoul, Korea, 2008, pp. 10 822–10 825.

[13] S. Kim, H. Seo, J. Shin, and H. J. Kim, “Cooperative aerial manipulation using multirotors with multi-DOF robotic arms,” *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 2, pp. 702–713, Apr. 2018.

[14] M. V. Minniti, F. Farshidian, R. Grandia, and M. Hutter, “Whole-body MPC for a dynamically stable mobile manipulator,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3687–3694, Oct. 2019.

[15] U. Nagarajan, G. Kantor, and R. L. Hollis, “Trajectory planning and control of an underactuated dynamically stable single spherical wheeled mobile robot,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, 2009, pp. 3743–3748.

[16] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, “Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results,” *Front. Robot. AI*, vol. 3, 2016, Art. no. 16.

[17] E. Simetti, G. Casalino, F. Wanderlingh, and M. Aicardi, “Task priority control of underwater intervention systems: Theory and applications,” *Ocean Eng.*, vol. 164, pp. 40–54, 2018.

[18] O. Kanoun, F. Lamiraux, and P.-B. Wieber, “Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task,” *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 785–792, Aug. 2011.

[19] M. Edmonds, J. Yi, N. K. Singa, and L. Wang, “Generation of high-density hyperspectral point clouds of crops with robotic multi-camera planning,” in *Proc. IEEE Conf. Autom. Sci. Eng.*, Vancouver, BC, Canada, 2019, pp. 1475–1480.

[20] A. Muetze and Y. C. Tan, “Electric bicycles - A performance evaluation,” *IEEE Ind. Appl. Mag.*, vol. 13, no. 4, pp. 12–21, Jul./Aug. 2007.

[21] A. Bechar and C. Vigneault, “Agricultural robots for field operations: Concepts and components,” *Biosyst. Eng.*, vol. 149, pp. 94–111, 2016.

[22] 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.*, Singapore, 2017, pp. 2414–2419.

[23] P. Wang, J. Yi, and T. Liu, “Stability and control of a rider–bicycle system: Analysis and experiments,” *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 348–360, Jan. 2020.

[24] Y. Zhang, P. Wang, J. Yi, D. Song, and T. Liu, “Stationary balance control of a bikebot,” in *Proc IEEE Int. Conf. Robot. Autom.*, 2014, pp. 6706–6711.

[25] P. Seekhao, K. Tungpimolrut, and M. Parnichkun, “Development and control of a bicycle robot based on steering and pendulum balancing,” *Mechatronics*, vol. 69, 2020, Art. no. 102386.

[26] L. Keo and M. Yamakita, “Control of an autonomous electric bicycle with both steering and balancer controls,” *Adv. Robot.*, vol. 25, no. 1/2, pp. 1–22, 2011.

[27] N. Takasugi, “Nonlinear balance control of an unmanned bicycle: Design and experiments,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2020, pp. 7279–7284.

[28] Y. Zhang, J. Li, J. Yi, and D. Song, “Balance control and analysis of stationary riderless motorcycles,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, 2011, pp. 3018–3023.

[29] S. Singhania, I. Kageyama, and V. M. Karanam, “Study on low-speed stability of a motorcycle,” *Appl. Sci.*, vol. 9, no. 11, 2019, Art. no. 2278.

[30] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed. Berlin, Germany: Springer, 2017.

[31] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC Press, 1994.

[32] Y. Gong, K. Chen, J. Yi, and T. Liu, “Control of a two-wheel steering bikebot for agile maneuvers,” in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, 2019, pp. 984–989.

[33] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*. Boca Raton, FL, USA: CRC Press, 2007.

[34] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Berlin, Germany: Springer, 2006.

[35] G. Antonelli, “Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems,” *IEEE Trans. Robot.*, vol. 25, no. 5, pp. 985–994, Oct. 2009.

[36] “User Guide, KINOVA Gen2 ultra lightweight robot,” Kinova Inc., Boisbriand QC, Canada, 2019. [Online]. Available: https://www.kinovarobotics.com/en/resources/gen2-technical-resources

[37] F. Han and J. Yi, “Stable learning-based tracking control of underactuated balance robots,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1543–1550, Apr. 2021.

Feng Han (Student Member, IEEE) received the B.S. degree in aerospace engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2017, and the M.S. degree in aerospace engineering from Harbin Institute of Technology, Harbin, China, in 2019. He is currently working toward the Ph.D. degree in mechanical and aerospace engineering with Rutgers University, Piscataway, NJ, USA.

His research interests include dynamics, control, and machine learning with the applications to robotics.

Alborz Jelvani is currently working toward the B.S. degree in computer science with Rutgers University, Piscataway, NJ, USA.

His research interests include mechatronics, embedded operating systems, distributed systems, and topics in theoretical computer science.

Jingang Yi (Senior Member, IEEE) received the B.S. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 1993, the M.Eng. degree in precision instruments from Tsinghua University, Beijing, China, in 1996, and the M.A. degree in mathematics and the Ph.D. degree in mechanical engineering from the University of California, Berkeley, CA, USA, in 2001 and 2002, respectively.

He is currently a Professor of Mechanical Engineering with Rutgers University. His research interests include autonomous robotic systems, dynamic systems and control, mechatronics, automation science and engineering, with applications to biomedical systems, civil infrastructure, and transportation systems.

Dr. Yi is a fellow of the American Society of Mechanical Engineers (ASME). He was the recipient of the 2010 US NSF CAREER Award. He currently serves as a Senior Editor for the IEEE ROBOTICS AND AUTOMATION LETTERS, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and as an Associate Editor for *International Journal of Intelligent Robotics and Applications*. He served as an Associate Editor for IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, IEEE/ASME TRANSACTIONS ON MECHATRONICS, IEEE ROBOTICS AND AUTOMATION LETTERS, *IFAC Journal Mechatronics*, *Control Engineering Practice*, and *ASME Journal of Dynamic Systems, Measurement and Control*.

Tao Liu (Senior Member, IEEE) received the B.S. degree in mechanical engineering from the Harbin University of Science and Technology, Harbin, China, in 2001, the M. Eng. degree in mechanical engineering from the Harbin Institute of Technology, Harbin, China, in 2003, and the D.Eng. from Kochi University of Technology, Kochi, Japan, in 2006.

He was an Assistant Professor with the Department of Intelligent Mechanical Systems Engineering, Kochi University of Technology, from 2009 to 2013. He is currently a Professor with the State Key Laboratory of Fluid Power Transmission and Control and School of Mechanical Engineering, Zhejiang University, Hangzhou, China. He is an Inventor of one Japanese patent about wearable sensors for gait analysis that has been successfully commercialized. His research interests include wearable sensor systems, rehabilitation robots, biomechanics, and human motion analysis.

Dr. Liu was the recipient of the Japan Society of Mechanical Engineers Encouragement Prize, in 2010.