Coordinated Pose Control of Mobile Manipulation with an Unstable Bikebot Platform

Feng Han, Alborz Jelvani, Jingang Yi, and Tao Liu

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 the 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 a plant inspection with end-effector position and orientation errors about 5 mm and 0.3 degs, respectively.

Index Terms—Underactuated robots, balance control, mobile manipulation, task priority planning, bicycle control

I. INTRODUCTION

Mobile manipulation integrates a mobile robot with an onboard multi-link 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, ocean waves and tides cause large dynamic disturbances to the ship-mounted manipulator [12] and a wheeled/legged mobile robot would fall down when moving on a steep or rocky field [13]. For aerial manipulation, interaction forces generates large disturbances for robot motion and control due to small masses and limited actuation of the quadrotors [7], [14].

This work was partially supported by the US National Science Foundation under award CNS-1932370.

F. Han and J. 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).

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

T. Liu is with the State Key Lab of Fluid Power and Mechatronic Systems and the School of Mechanical Engineering, Zhejiang University, Hangzhou, Zhejiang 310027 China (email: liutao@zju.edu.cn).

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 [15], a model predictive control was presented for collaborative manipulation, balancing and interaction of a ball-based three degree-of-freedom (DOF) manipulator. In [16], an unstable mobile manipulation used a single spherical wheel-based “ballbot” as the mobile platform and a balance motion control was developed for the underactuated, nonholonomic robot. For kinematic redundant manipulators, task-priority control takes advantages of design space in the null space of the Jacobian matrix. Velocity control was designed through optimization to satisfy the control tasks from the highest to lowest priorities [17]–[19].

In this paper, we present a mobile manipulation system that is built on an autonomous bikebot. A 6-DOF lightweight manipulator is mounted on the bikebot and the system was developed for agricultural applications such as autonomous plant inspection and scouting [20]. All existing agricultural robots are built on double-track platform and their energy consumption is much higher than that of single-track mobile robots such as bikebot [21]. 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 avoids potential severe soil compaction [22]. Steering and speed control of autonomous one-wheel steered bikebot has been reported (e.g., [23], [24]) 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 [25]–[28]. However, the additional balance actuators increase the system complexity and operation cost. In this work, steering is used as the only actuation for balancing the stationary platform.

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) [29], [30]. It is desirable to design new mechanisms and control methods to enlarge the controllable roll motion range for practical bikebot 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 task. Therefore, robotic applications require the stationary balance capability of single-track mobile robots.
This paper presents the coordinated control of the bikebot and the manipulator to enhance the stationary balance and posture control task. 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, double-track Segway-like balance robots. Second, the proposed coordinated motion control design 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 remainder of this paper 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 before we summarize the concluding remarks in Section VI.

II. PROBLEM STATEMENT AND SYSTEMS DYNAMICS

In this section, we first present the problem statement and then the dynamic models of the bikebot manipulation system.

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 l. Two sets of coordinate frames are introduced: an inertial frame Ω and body frame F_i for the ith manipulator link, i = 1, · · ·, n. Frame F_i is constructed by following the DH parameter convention [31]. F_0 and 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 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 Θ = [θ_1 · · · θ_n]^T as the manipulator joint angles, i = 1, · · ·, n. The generalized coordinates of the system are denoted as q = [φ_b Θ]^T ∈ Q ⊂ R^{n+1}, where Q is admissible set for q. We denote the pose (i.e., position and orientation) of end-effector E in Ω as ξ_e ∈ R^6.

Problem Statement: Given a set of Nξ desired poses {ξ_e^k}_{k=1}^Nξ, Nξ ∈ 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 ξ_e^k, k = 1, · · ·, Nξ.

B. Systems Dynamics

We use DH parameters (θ_i, d_i, a_i, α_i) for the ith link of the manipulator, i = 1, · · ·, n. The homogeneous transformation matrix from F_i to F_0 is written as [32]

\[ T_i(q) = A_i^1 A_i^2 \cdots A_i^{i-1}, \]

where A_i^{i-1} denotes the transformation from F_i to F_{i-1} as

\[ A_i^{i-1} = \begin{bmatrix} R_i^{i-1} & p_i \[0 \times 1_3 \end{bmatrix}, \]

\[ R_i^{i-1} = R_z(\theta_i) R_x(a_i), \quad R_j(\beta) ∈ SO(3), j = x, y, z, \]

\[ p_i = [a_i \cos \theta_i \ a_i \sin \theta_i \ d_i]^T \]

is the corresponding position vector in F_{i-1}. With (1), we write the pose of end-effector E in F_0 as ξ_e^0 = ξ_e(T_n(q)).
We denote the mass center position of the $i$th link in $\mathcal{F}_0$ as $p_i^0$, and its position in $\mathcal{I}$ is $p_i = \mathbf{R}_i^0(p_0 + p_i^0)$, where $p_0$ is the position vector of point $S$ in $\mathcal{I}$. The linear velocity $\mathbf{v}_i^0$ and angular velocity $\omega_i^0$ of the $i$th link in $\mathcal{F}_0$ are obtained as

$$\dot{x}_{\mathcal{F}_0} = \begin{bmatrix} (\mathbf{v}_i^0)^T & (\omega_i^0)^T \end{bmatrix}^T = J_i \hat{\phi},$$  

where $J_{\phi i} \in \mathbb{R}^{6 \times n}$ is the Jacobian from $\mathcal{F}_i$ and $\mathcal{F}_0$. Therefore, the linear velocity $\mathbf{v}_i$, and angular velocity $\omega_i$, in $\mathcal{I}$ are

$$\mathbf{v}_i = \omega_b \times \mathbf{R}_i^T(p_0 + p_i^0) + \mathbf{R}_i^0 \mathbf{v}_i^0, \quad \omega_i = \omega_b + \mathbf{R}_i^0 \omega_i^0,$$

where $\omega_b = [\phi_b 0 0]^T$ is the platform's roll velocity.

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

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

where $T_b = \frac{1}{2} \omega_b^T I_b \omega_b + \frac{1}{2} m_i v_i^T v_i$ is the kinetic energy for the bikebot and for the $i$th link of the manipulator $T_i = \frac{1}{2} m_i v_i^T v_i + \frac{1}{2} \omega_i^T R_i^T I_i(R_i^T)^T \omega_i$, $m_i$ and $m_i$ are respectively the masses for the bikebot and the $i$th link, $\delta_{TC}$ is the velocity of the mass center $G$, $I_b = \text{diag}(I_b, 0)$, and $I_i$ are the inertia matrices for the bikebot and the $i$th link about their mass center, respectively. For potential energy terms in (5), for the bikebot, $U_b = m_i g (p_G \cdot e_z + \Delta h_G)$ and for the $i$th link, $U_i = m_i g p_i^0 \cdot e_z$, where $p_G$ is the position vector of $G$ in $\mathcal{I}$, $g = 9.8$ m/s$^2$ is the gravitational constant, unit vector $e_z = [0 0 1]^T$, and $\Delta h_G$ is the height change of $G$ due to steering actuation.

The dynamic model is obtained by Lagrangian method as

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

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

$$\begin{bmatrix} D_{bb} & D_{bi} \\ D_{bi} & D_{ii} \end{bmatrix} \begin{bmatrix} \dot{q}_b \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} C_b \\ C_{\phi} \end{bmatrix} \dot{q} + \begin{bmatrix} G_b \\ G_{\phi} \end{bmatrix} = \begin{bmatrix} \tau_b \\ \tau_{\phi} \end{bmatrix},$$

where the block matrices are in appropriate dimensions and their dependencies on $q$ and $\dot{q}$ are dropped for presentation brevity. We will derive the steering-induced torque model for $\tau_b$ 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 $\phi$ is built on an initial steering angle $\phi_0$ with a small increment $\delta$, namely, $\phi = \phi_0 + \delta$. Increment $\delta$ is small and the position change of point $G$ under $\phi$ is negligible.

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

$$R_{\mathcal{F}_O}^T = R_{\phi}(\varepsilon)R_{x}(\phi_b)R_{z}(\phi).$$

We denote the angle between the wheel plane and the ground as $\gamma$ and it is straightforward to obtain

$$\cos \gamma = R_{\mathcal{F}_O}^T e_y \cdot e_z = \sin \phi \sin \varepsilon - \cos \phi \cos \varepsilon \sin \phi_b,$$

where unit vector $e_y = [0 1 0]^T$.

To simplify the two-wheel steering design, both the front and rear steering angles, denoted respectively by $\phi^f$ and $\phi^r$, are controlled and kept at symmetric position (i.e., same amplitude but opposite directions) for all time, namely, $\phi^f = -\phi^r = \phi$, with $\phi^f = \phi_f + \delta$, $\phi^r = \phi_r - \delta$, and $\phi_f + \phi_r = \phi_0$. For brevity, we only use $\phi$ and $\delta$ in the following derivation. Under small $\delta$, wheel/groun contact points $C_1$ and $C_2$ move to $C_1'$ and $C_2'$, respectively. Fig. 2(b) illustrates the wheel/gound contact points under $\delta$. 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 $O_G'$ ($O_b'$), projected points $O_f^T (O_b^T)$ on the ground. The radii of the circular arc $C_1C_1'$ ($C_2C_2'$) and the bikebot wheel are denoted as $r$ and $R$, respectively. From the geometric relationship, we obtain

$$r = R \cos \gamma = R \sin \phi \sin \varepsilon - R \cos \phi \cos \varepsilon \sin \phi_b.$$

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 $\phi_{ob} = \angle AO_b'x_{ob}$ and $\phi_{bg} = \angle AO_b'x_{bg}$ (for the rear wheel). Let $L$ denote the distance from the $C_1C_2$ to $O_f^T O_b^T$ and $L$ is obtained by the geometry relationship as

$$L = r \cos \phi_b.$$
The relationship between $\phi_g$ and $\phi$ is captured by \[ \phi_g = \arctan \left( \frac{\cos \varepsilon}{\cos \varphi_b \tan \phi} \right). \] (11)

Given a fixed $\phi_0$, 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 $\varphi_b$, $\phi_0$ and $\delta$.

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 $C'_1C'_2$ and obtain
\[ \tau_b \approx m_{bg} L = m g L (\varphi_b, \phi_0, \delta). \] (12)

It is helpful to find steering angle $\phi_0$ at which the increment $\delta$ generates largest torque increase of $\tau_b$. Thus, we introduce the steering torque sensitivity as the steering torque rate with respect to $\delta$ at $\varphi_b = 0$, namely,
\[ S_\tau (\phi_0) = \left. \frac{\partial \tau_b}{\partial \delta} \right|_{\varphi_b=0} = m_{bg} r_{\phi_0} \frac{\cos^2 \varepsilon \tan \phi_0 (tan^2 \phi_0 + 1)}{(\cos^2 \varepsilon \tan^2 \phi_0 + 1)^{3/2}}. \]

From above equation, it is straightforward to see 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 $\phi_0^* = \frac{\pi}{2}$ rad, $S_\tau (\phi_0)$ reaches its maximum value as
\[ S_\tau (\phi_0^*) = m_{bg} r_{\phi_0^*} \frac{1}{\sqrt{\cos^2 \varepsilon}} = m_{bg} g R \tan \varepsilon. \]

Therefore, we focus on using $\phi_0^* = \frac{\pi}{2}$ rad 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 then calculated as
\[ \tau_{90} = m_{bg} r_{90} \cos \phi_g = -m_{bg} g R \sin \varepsilon \cos \delta \sin \left( \frac{\delta}{\cos \varepsilon} \right), \] (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 larger caster angle configuration helps increase the steering-induced balance torque and therefore improve balance capability. For any other initial steering angle $\phi_0$, the radius is calculated by (9) and the steering torque $\tau_b$ 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 initial steering angle $\phi_0 = 0$. Indeed, the configuration is similar to double-track balance robot such as Segway. This observation explains and 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 the bikebot balance implementation.

IV. Coordinated Balance Control Design

In this section, we focus on the coordinated balance and pose control of the bikebot manipulation. The controller design is built on the balancing equilibrium manifold (BEM).

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 $\varphi_b$ and manipulator joint angles $\Theta$ should satisfy an intrinsic relationship that is captured by BEM. From (6), the equation of motion of the bikebot is written as
\[ D_{bb} \dot{\varphi}_b + D_{bg} \dot{\Theta} + C_b \ddot{q} + G_b(q) = \tau_b \] (14)
where $G_b(q)$ is the total gravitational torque from the bikebot and the manipulator. Considering the quasi-static motion, namely, $\ddot{q} = 0$, we define the BEM as
\[ \mathcal{E} = \{ q_e = [\varphi_b^T \Theta^T]^T : G_b(q) = \tau_b, q \in \mathbb{Q} \}. \] (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 $\tau_b^{\max}$ with possible $\Theta$.

To move the end-effector from one pose to another, a trajectory should be designed around the BEM at any time, namely, $q \in \mathcal{E}$. A velocity constraint should be enforced given the BEM and limited steering actuation. Using (13), the steering torque is $\tau_b = -M g R \sin \varepsilon \cos \delta \sin \delta_g$, where $M = m_{bg} + \sum_{i=1}^{n} m_i$ is the total mass of the entire system. Taking derivative of BEM condition $G_b(q) = \tau_b$, we obtain
\[ \dot{G}_b = \frac{\partial G_b}{\partial q} \dot{q} = -M g R \sin \varepsilon \frac{d}{d\delta} (\cos \delta \sin \delta_g) \delta = h(\delta) \dot{\delta}, \]
where $h(\delta) = -M g R \sin \varepsilon \frac{d}{d\delta} (\cos \delta \sin \delta_g)$. Defining $J_G = \frac{\partial G_b}{\partial q}$ as a Jacobian-like matrix, the above velocity constraint is specified as
\[ |J_G \dot{q}| \leq h_{\max} \dot{\delta}_{\max}, \] (16)
where $h_{\max} = \sup |h(\delta)|$ and $\dot{\delta}_{\max}$ is the maximum steering angular rate. Constraint (16) implies that when designing the trajectory $q(t)$, 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
\[ \mathcal{X}(q) = \{ \xi_e : \xi_e = \xi_e(T_{n+1}(q)), q \in \mathcal{E}, |\tau_b| \leq \tau_b^{\max} \}, \] (17)
where $T_{n+1}(q)$ is the homogeneous transformation from $F_n$ to $\mathcal{I}$. We further define the local end-effector pose workspace $\mathcal{X}_{\varphi_b}^{\phi_0}(\Theta) \subseteq \mathcal{X}(q)$ under roll angle $\varphi_b^0$, $q^0 = [\varphi_b^0 \Theta^T]^T$,
\[ \mathcal{X}_{\varphi_b}^{\phi_0}(\Theta) = \{ \xi_e : \xi_e = \xi_e(T_{n+1}(q^0)), q^0 \in \mathcal{E} \}. \]

The rationale to introduce $\mathcal{X}_{\varphi_b}^{\phi_0}(\Theta)$ is to specify the bikebot roll angle $\varphi_b^0$ 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 $\xi^k$, $k = 1, \ldots, N$, the BPIK is to find optimal bikebot roll and manipulator joint angles $q^*_k$ as

$$ q^*_k = \arg\min_{q} \lambda_1 \Gamma_1 + \lambda_2 \Gamma_2 + \lambda_3 \Gamma_3 \quad \text{Subj. to } \lambda_4 |G_b| \leq e^{\text{max}}, \ q \in \mathcal{E}, $$

where \( \Gamma_1 = ||\xi^k - \xi_e(T_{n+1}(q))||^2 \), \( \Gamma_2 = |G_b(q^*_k) - G_b(q_k^{* - 1})|^2 \), \( \Gamma_3 = e_k^T P e_k - q^* - q_k^{* - 1} \), \( \lambda_1, \lambda_2, \lambda_3 > 0 \), and \( \lambda_4 > 1 \) are weight parameters, \( P > 0 \) is a symmetric positive definite matrix. We initialize with \( q_k^{* - 1} = 0 \) and \( G_b(q_k^{* - 1}) = 0 \).

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., \( \Gamma_1 \)). 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 \( \Gamma_2 \) and \( \Gamma_3 \) in (18a) try to minimize the difference between configurations at the current and the previous steps. We use the BPIK to obtain \( q^*_k \) from \( \xi^k \). We first search the solution in local workspace \( X(q) \) to avoid large bikebot movement. If this is impossible, the BPIK then searches the solution in the workspace \( X^*(q) \). If the calculated feasible poses are outside of \( X(q) \), (18) will return the closest results. Once obtaining the desired joint angles \( \{q^*_k\}_{k=1}^N \), we need to design transition trajectory along \( \mathcal{E} \) between each two consecutive poses.

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

$$ \min_{q^*(t)} \int_{t_0}^{t_f} e_k^T W_1 e_k + q^T W_2 q + (\delta G_b, k)^2 dt \quad \text{(19a)} $$

Subj. to \( \dot{q}(t_0) = q_k^{* - 1}, \dot{q}(t_f) = q_k^* \), \( \ddot{q}(t_0) = \ddot{q}_k^{* - 1}, \ddot{q}(t_f) = \ddot{q}_k^* \), \( D_{\theta} \dot{q} + D_{\theta} \ddot{\theta} + C_0 q + G_\theta = \tau_\theta \), \( |J_C q| \leq h_{\max} \xi_{\max} \lambda_4 |G_b| \leq \tau_{\theta, \max} \), \( |\tau_{\theta, i}| \leq \xi_{\max}, q \in \mathcal{Q}, |\dot{q}| \leq \dot{q}_{\max}, |\ddot{q}| \leq \ddot{q}_{\max} \),

where \( \delta G_b, k = G_b(q) - G_b(q_k^{* - 1}) \), \( W_1, W_2 \in \mathbb{R}^{n+1} \) are positive diagonal matrices, and \( \xi_{\max} \) is the maximum joint torque of the \( i \)-th link, \( i = 1, \ldots, n \). To consider the quasi-static motion, the angular velocity and acceleration of the manipulator are bounded as in (19c). The constraint in (19d) is similar to that in (18c) 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 solution in each dimension of \( q(t) \). We use Bézier polynomial because of its attractive properties [34]. The solution \( q(t) \) is written in term of \( N \)th order Bézier polynomials (\( N \in \mathbb{N} \)) as

$$ \varphi_b = b(s, p_b) = \sum_{j=0}^{N} p_{b_j} b_j(s), \end{equation}$$

for \( i = 1, \ldots, n \), where \( b_j(s) = \frac{N!}{(N-j)!j!} (1-s)^{N-j} s^j \), parameters \( p_b = [p_{b_0} \cdots p_{b_N}]^T \) and \( p_{b_0} = [p_{b_0} \cdots p_{b_N}]^T \). 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 \( q(t_0) = b(0, p_b) b(0, p_{b_0}) \cdots b(0, p_{b_N})^T \), \( q(t_f) = b(1, p_b) b(1, p_{b_0}) \cdots b(1, p_{b_N})^T \) and \( q(t) \) is then written as polynomials of \( s \) with parameters \( p = \{p_b, p_{b_1}, \ldots, p_{b_N}\} \).

For \( q(t) \), we obtain \( \varphi_b = \sum_{j=0}^{N} p_{b_j} b_j(s) \int_0^s dt \) and \( \theta_i = \int_0^{N} p_{b_j} b_j(s) \int_0^s dt \), \( i = 1, \ldots, n \). Noting that \( \int_0^1 \frac{dt}{\tau_{\theta, \max}} = \frac{1}{\tau_{\theta, \max}} \) is constant, \( \varphi(t) \) and \( \dot{\varphi}(t) \) are written in term of \( p \). 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 \( p \).

To solve the optimization problem, 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 \( q^*(t) \) [35]. In each iteration, a total of \( 3N_s(n+1) \) evaluations of \( b(s, p) \) for \( q, \dot{q}, \ddot{q} \) 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 \( O(N^2 n^2) \). Therefore, the computational complexity for solving (19) by the proposed Bézier polynomial-based approach is \( O((N + N_s)N_s n^2) \). As a comparison, a dynamic programming (DP) method can be used to solve (19) with complexity \( O(N_s^2 n^2) \). 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 \( q^*(t) \) for a given set of end-effector poses \( \{\xi^k\}_{k=1}^N \). In this subsection, we present the online control algorithms to follow the trajectory \( q^*(t) \).
where $M_b > 0$ is a finite constant that is related to $e_b(0)$ and $\dot{e}_b(0)$. Therefore, for any $t \geq t_0 := \frac{\sqrt{2e_b(0)}}{\lambda_p}$, $|e_b(t)| \leq e_b$. To show the tracking error convergence for $e_{\Theta}$, we consider a Lyapunov function candidate $V(t) = e_{\Theta}^T e_{\Theta} = \|e_{\Theta}\|^2 > 0$ for any non-zero error $e_{\Theta}$. Let $l_0$ denote the upper bound for input $l_0 \delta_{\Theta}$ during $t \in [0, t_0]$, namely, $l_0 := \sup_{0 \leq t \leq t_0} \|l_0 \delta_{\Theta}\|$. Moreover, we have

\[
\dot{V}(t) = -2 e_{\Theta}^T K_p e_{\Theta} + 2 l_0 e_{\Theta}^T \delta_{\Theta} \leq -2 \lambda_p \|e_{\Theta}\|^2 + 2 l_0 \|e_{\Theta}\|,
\]

where $\lambda_p = \min_{1 \leq i \leq n} K_{pi}$. We introduce $W(t) = \sqrt{V(t)} = \|e_{\Theta}(t)\|$ from above inequality, we obtain $W(t) \leq -\lambda_p W(t) + l_0$. Thus, we have $\frac{d}{dt}(W e^{\lambda_p t}) = W e^{\lambda_p t} + W\lambda_p e^{\lambda_p t} \leq l_0 e^{\lambda_p t}$ and integrating from 0 to $t$, we obtain

\[
W(t)e^{\lambda_p t} - W(0) \leq \frac{l_0}{\lambda_p}(e^{\lambda_p t} - 1).
\]

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

\[
\begin{align*}
\|e_{\Theta}(t)\| &\leq \|e_{\Theta}(0)\|e^{-\lambda_p t} + \frac{l_0}{\lambda_p}(1 - e^{-\lambda_p t}) \\
&\leq \|e_{\Theta}(0)\|e^{-\lambda_p t} + \frac{l_0}{\lambda_p}.
\end{align*}
\]

(24)

From the above analysis, $\dot{V}$ is negative outside of the compact set $S = \{ e_{\Theta} : \|e_{\Theta}\| \leq \frac{l_0}{\lambda_p} \}$ and $e_{\Theta}$ exponentially converges to $S$. Note that for $t \geq t_0, \dot{t}_0 = 0$, $e_{\Theta}(t)$ converges to zeros exponentially due to stable dynamics (21).

With error bound in (23) and (24), we obtain the error bound for $e_q = q - q^* = [e_{\Theta} e_{q}^T]^T$ as

\[
\begin{align*}
\|e_q(t)\| &\leq \sqrt{e_{\Theta}^2(t) + \|e_{\Theta}(t)\|^2} \leq |e_b(t)| + \|e_{\Theta}(t)\| \\
&\leq M_b e^{-\lambda_p t} + \|e_{\Theta}(0)\|e^{-\lambda_p t} + \frac{l_0}{\lambda_p}.
\end{align*}
\]

(25)

Considering pose error $e_{\xi} = \xi_{\text{e}}(q) - \xi_{\text{e}}(q^*)$ and $q = q^* + e_q$, we have

\[
\begin{align*}
e_{\xi} &= \xi_{\text{e}}(q^*) + \frac{\partial \xi_{\text{e}}}{\partial q} q_e + \Delta_q - \xi_{\text{e}}(q^*) = J_q(q^*) e_q + \Delta_q,
\end{align*}
\]

(26)

where $J_q(q) \in \mathbb{R}^{6 \times (n+1)}$ is the Jacobian matrix from $F_n$ (end-effector $E$) to inertial frame $\mathcal{I}$ and $\Delta_q \in \mathbb{R}^6$ is the higher order term of error $e_q$. From (26), it is straightforward to obtain that $\|e_{\xi}(t)\| \leq \|J_q(q^*)\| \|e_q(t)\| + \|\Delta_q\|$. For the higher order term $\Delta_q = O(\|e_q(t)\|^2)$, there exists a finite constant $M_\delta > 0$ such that $\|\Delta_q\| \leq M_\delta \|e_q(t)\|$ due to (25) and then $\|e_{\xi}(t)\| \leq M_q \|e_q(t)\|$, where $M_p = \sup_{q^*} \|J_q(q^*)\|^2 + M_\delta$. Therefore, the pose error $e_{\xi}(t)$ 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 6-DOF robotic manipulator (Jaco2 from Kinova Inc.). Fig. 4 illustrates the interconnection schematic of the embedded systems and actuators and sensors. Both the 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.

To validate the steering mechanism and models, we also built and conducted experiments to measure the tire/ground contacts and movement. Fig. 5 shows the experimental setup for validation of the steering mechanism and balance torque model. Table II lists the model parameter values for the bikebot. The values of the DH parameters and mass moments of inertia of the manipulator links are listed in Table II. The other physical parameters for each link can be found in [37].

![Fig. 5. Experimental setup for validation of the steering mechanism and balance torque model.](image)

**TABLE II**

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

**B. Experimental Results**

We first present the validation of the steering balance models. Figs. 6(a) and 6(b) show the values of the turning radius $r_{\phi_0}$ and the steering torque sensitivity $S_T$, respectively, as the static steering angle $\phi_0$ 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 $S_T$ also reaches its maximum point around $\phi_0 = 90$ degs with $S_T = 0.87$ Nm/deg. At $\phi_0 = 0$, the projected radius $r_0$ and steering torque sensitivity $S_T$ are near zero. From this observation, an initial steering angle $\phi_0$ is chosen around 90 degs for following stationary balance experiments. At $\phi_0 = 90$ degs, multiple stationary balancing experiments were conducted. Fig. 6(c) shows the steering-induced balance torque $\tau_\delta$ at different roll angles $\phi_0$ and increments $\delta$. Multiple experimental trials are plotted together with the steering torque model prediction from [13], i.e., the 3D surface as shown in the
The experimental data are scattered around the torque model prediction with small errors. These results validate the steering-induced balance torque model.

Fig. 8. Verification of the recoverable roll angle region. The markers “x”, “*” and “o” 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 $\varphi_b = \pm 3.4\degree$, $\pm 5.6\degree$, and $\pm 11.6\degree$.

Fig. 7 shows the bikebot balance control results. 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 back and forth around the 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 at $t = 270\,s$, multiple disturbances were applied by manually pushing 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 actual 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 predictions, we conducted multiple balance control experiments. One experimental trial was considered successful if the system was kept balance for a time period over 50 s. Fig. 7 shows the successful trials in the $\varphi_b-\varphi_\theta$ plane. Each marker in the figure represents the state at which the bikebot successfully started to balance under steering control, which was confirmed by comparing the model predictions from (13) and dynamics (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.

**TABLE III**

| Pose | Desired $\xi_b^k$ | BPIK-planned $\xi_b(T_{n+1}(q_n^k))$ | Actual $\xi_b^k$ | Errors |
|------|------------------|-------------------------------|------------------|--------|
| 1    | $[-13 - 55 94 - 61 52 96]^{T}$ | $[-14 - 56 95 - 64 52 98]^{T}$ | $[-14 - 56 95 - 63 51 97]^{T}$ | $0.76 \pm 0.02$ $0.43 \pm 0.005$ |
| 2    | $[-15 - 67 109 57 31 - 57]^{T}$ | $[-15 - 65 106 58 30 - 58]^{T}$ | $[-15 - 67 109 53 30 - 58]^{T}$ | $0.34 \pm 0.01$ $0.16 \pm 0.017$ |
| 3    | $[-17 - 69 106 38 55 - 37]^{T}$ | $[-16 - 68 107 37 55 - 37]^{T}$ | $[-16 - 68 107 35 55 - 37]^{T}$ | $0.72 \pm 0.13$ $0.40 \pm 0.08$ |
| 4    | $[-15 - 58 114 32 48 - 21]^{T}$ | $[-15 - 61 112 39 42 - 18]^{T}$ | $[-15 - 60 112 39 42 - 18]^{T}$ | $0.34 \pm 0.09$ $0.15 \pm 0.046$ |

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.
We now present a plant inspection example for the end-effector to continuously go through and stop momentarily at four poses (i.e., $N = 4$). This represents the end-effector movement during a plant scanning and inspection task [20]. 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 $I$ and orientation always pointed towards 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 3D trajectories with the four poses. The planning and control parameters used in experiments include: $\lambda_1 = 10$, $\lambda_2 = 1$, $\lambda_3 = 5$, $\lambda_4 = 1.5$, $W_1 = \text{diag}(10, 5, 5, 5, 1, 1, 1)$, $W_2 = \text{diag}(1, 1, 1, 1, 1, 1)$, $\kappa = 5$, $\epsilon = 0.1$, $\epsilon_0 = 0.4$ degs, $\dot{q}_{\text{max}} = 36$ deg/s, $\ddot{q}_{\text{max}} = 120$ deg/s$^2$, $\tau_{\text{max}} = [10\ 15\ 10\ 5\ 5\ 5]^T$ Nm, $\delta_{\text{max}} = 15$ degs, $\dot{\delta}_{\text{max}} = 20$ deg/s, and the degree of the Bézier polynomial $N = 7$. For off-line planning implementation, the number of data points was chosen as $N_s = 50$ in each dimension of $q(t)$ 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 the real-time motion control of the current pose movement. By doing so,
the proposed planner was capable of obtaining the trajectory with fast computational time. Table III lists the desired end-effector poses, the poses planned by the BPIK and the actual poses under the controller. From the table, 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. Figs. 10(a) and 10(b) show the bikebot roll angle $\phi_b$ and the six joint angles $\Theta$ of the manipulator, respectively. Since $\phi_b$ and the first three joint angles ($\theta_1$-$\theta_3$) played a major role to balance the entire system, their reference trajectories were designed to avoid large variations in the figures. Poses 2 and 3 were searched in the local workspace $X_{\text{wp}}(\Theta)$ of Pose 1, and Pose 4 is searched in the workspace $X(q)$. 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).

C. Discussion

To further examine the performance, we conducted additional comparison experiments. Fig. 10(a) shows the bikebot roll angle when the bikebot balance priority is not considered in the trajectory planning. Clearly, the entire system lost balance in the pose transition phase at $t = 50$s. The result confirms the challenge in manipulator pose movement control. We further conducted and repeated the 4-pose control experiment 15 times. Fig. 9(c) summarizes the end-effector pose errors statistics (i.e., mean and one-standard deviation) with and without online velocity correction under the proposed trajectory planning method as modeling errors and system uncertainties might exist. With the online modification 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. 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). Without the velocity correction, both the position and orientation tracking errors are larger than these with correction. The results demonstrate the successful balance and pose control performance by the design. We also conducted computational time comparison between the proposed trajectory planning algorithm and the DP method and the result is shown in Table IV. The numerical results confirmed that the computational 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 in this work 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 accurate model of the system and it is desirable to extend to handle model parameter uncertainties in complex, dynamic environment. One possible approach is to use machine learning-based methods. For example, as discussed in [38], the robot dynamics might be approximated and estimated using a Gaussian process learning approach and a learning-based motion control can be then designed.

VI. Conclusion

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 first 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 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 extensive 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. 2012 IEEE Symp. Robot. Appl., Kuala Lumpur, Malaysia, 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,” Rob. Auton. Syst., vol. 122, pp. 1–12, 2019.

[4] B. Arad, J. Balendonck, R. Barth, O. Ben-Shahar, Y. Edan, T. Helleström, J. Hemming, P. Kurtsner, O. Ringdahl, T. Tielens, and B. van Tuyl, “Development of a sweet pepper harvesting robot,” J. Field Robotics, 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, 2015.
