Balancing Control and Pose Optimization for Wheel-legged Robots Navigating Uneven Terrains

Junheng Li, Junchao Ma, and Quan Nguyen

Abstract—In this paper, we propose a novel approach on controlling wheel-legged quadrupedal robots using pose optimization and force control via quadratic programming (QP). Our method allows the robot to leverage wheel torques to navigate the terrain while keeping the wheel traction and balancing the robot body. In details, we present a rigid body dynamics with wheels that can be used for real-time balancing control of wheel-legged robots. In addition, we introduce an effective pose optimization method for wheel-legged robot’s locomotion over uneven terrains with ramps and stairs. The pose optimization utilized a nonlinear programming (NLP) solver to solve for the optimal poses in terms of joint positions based on kinematic and contact constraints during a stair-climbing task with rolling wheels. In simulation, our approach has successfully validated for the problem of a wheel-legged robot climbing up a 0.34 m stair with a slope angle of 80° and shown its versatility in multiple-stair climbing with varied stair runs and rises with wheel traction. Experimental validation on the real robot demonstrated the capability of climbing up on a 0.25 m stair with a slope angle of 30°.

I. INTRODUCTION

The recent technological and theoretical developments in both robot design and controls has allowed the world to witness many successful and highly-autonomous legged robots. With such hardware and software advancements, researchers in the robotics field are now facing a challenge to develop mobile legged robots that can conduct given tasks fully-autonomously and that the control framework can perform robustly in terrains with uneven surfaces with obstacles [1]. Glancing over the development of legged robots in the last decade, many bipedal and quadruped robots have demonstrated outstanding maneuverability and dynamic locomotion in unknown terrain and have proven to have great potential to be controlled autonomously (e.g. ATRIAS [2], Cassie [3], MIT Cheetah 3 [4], ANYmal quadruped [5], Big Dog [6], and Boston Dynamics Spot autonomous exploration mission [7]) However, energy efficiency in the robot hardware remains one of the most important condition that determines whether a mobile robot can perform real-life task that require extended period of time while maintaining highly dynamical locomotion, such as rescue and disaster-response missions [8]. Legged robots rely on gait sequence and proper foot placement to overcome obstacles and uneven surfaces, which is a more effective method in rough terrain locomotion compared to only wheeled system [9], while wheeled systems generally have far less energy consumption and faster speed to maneuver on an even surface [10]. Hence, the hybrid system of both wheels and legs could leverage the advantages from both worlds, enabling maneuverability in rough terrain, energy efficiency, and speed (e.g. [11] and [12]).

A. Related Work

Many successful research works on control of wheel-legged robots focus on connecting and coupling the wheel system and legged system. The wheel-legged quadruped ANYmal [13] utilizes a kinodynamics model in whole-body model-predictive control for robust locomotion control. The wheel-legged bipedal robot Ascento [14] adopts the whole-body dynamics by using linear-quadratic regulator (LQR) and Zero-moment Point (ZMP) in balance control. Our approach in controlling wheel-legged robots highly leverages the wheel traction to traverse challenging terrains (i.e., instead of stepping over a high obstacle, we use wheels to roll over the obstacle). This work also includes pose optimization to solve for optimal and collision-free poses for the robot climbing high stairs or obstacles. Force-based QP balancing control has successful implementations on quadruped robot MIT Cheetah 3 [4] and allows the robot to balance even after performing very dynamical and aerial tasks such as jumping [15]. The idea of modifying simplified dynamics in QP balance controller has also gain success in controlling mobile legged robot in the past (e.g. [16]). In our work, we take wheel dynamics into consideration and modify the simplified rigid body dynamics to connect the decoupled wheel and limb joint control in the hybrid wheel-legged system.

Trajectory optimization frameworks have allowed legged robots to achieve highly-dynamic locomotion. For example,
[17], [15], [18], and [19] all utilize NLP solver to find optimal trajectories for a certain task. One important constraint in such optimization frameworks is the utilization of robot dynamics, either full body dynamics or simplified centroidal dynamics. In [20], a NLP parameter optimization is used to find the optimal driving force in achieving highly dynamic locomotion. Our approach in pose optimization uses kinematic constraints instead to guarantee collision-free with the terrain and solve for favorable configurations to maintain balance. The optimal pose will then be used in combination with a balance controller using QP-based force control to maintain balance and desired pitch angle. Pose optimization also happens to resonate with the crawling mode introduced in [21] and [22], in which the crawling mode in rolling is proven to be a superior option in slope-climbing. Thanks to only few critical poses are needed, the computation intensity is dramatically scaled down compared to full trajectory optimizations. Unlike the approach in [12] by adapting wheel-legged robot’s posture in rough terrain with feedback control, or the approach in [23] by adding passive suspension for pose adapting, our approach of finding the optimal poses or robot configurations based on the terrain map.

B. Contributions

The main contributions of the paper are as follows:

- We introduce a rigid body dynamics with wheels that can be effectively used for balancing control of wheel-legged robots.
- We present a pose optimization method with kinematic and contact constraints to solve for optimal poses during a stair climbing tasks while maintaining wheel traction with the terrain.
- We propose a hybrid control framework that includes force-based QP and joint PD control to track optimal poses in order to achieve stable locomotion of wheel-legged robots navigating uneven terrains with stairs and ramps.
- The pose optimization is very efficient due to its small problem size. The solved optimal poses at a certain location can be linearly interpolated to obtain the joint trajectory at any given time during the task.
- The proposed approach allows our wheel-legged robot to climb up on a 0.34 m stair with a slope angle of 80° with wheel rolling traction during the entire motion. The hardware implementation has achieved stair and ramp climbing with the height of 0.25 m and ramp slope of 30°.

The rest of the paper is organized as follows. Section II introduces the wheel-legged robot model, its physical parameters. The overview of our control architecture is highlighted in Section III. Section IV presents the rigid body dynamics with wheels used in balancing control for wheel-legged robots, force-based QP control formulation, and rolling control. Section V introduces the pose optimization and its problem formulations, constraints, and pose tracking controller. Section VI highlights some simulation and experiment results with the proposed approach.

II. ROBOT MODEL AND HARDWARE

In this section, we introduce the model of our wheel-legged robot and its physical parameters. Fig. 1 presents the hardware assembly of the wheel-legged robot in motion. Our wheel-legged robot consists of a robot trunk, four sets of 3 degree of freedom(DoF) legs. Each leg consists of thigh, calf, and wheel. In total, the wheel-legged robot has 12 actuators. Fig. 2 shows the leg assembly and joint definitions. Table I includes all physical parameters of the robot. Our robot is built from the Unitree A1 quadruped robot. Parameters of the Unitree A1 actuator are shown in Table II.

![Fig. 2: Leg Configuration. The link and joint configuration of wheel-legged robot leg, rendered in SolidWorks.](image)

| TABLE I: Robot Physical Parameters |
|------------------------------------|
| Parameter | Symbol | Value | Units |
| Mass | \( m \) | 11.84 | kg |
| Body Inertia | \( I_{xx} \) | 0.0214 | kg \cdot m^2 |
| | \( I_{yy} \) | 0.0553 | kg \cdot m^2 |
| | \( I_{zz} \) | 0.0443 | kg \cdot m^2 |
| Body Length | \( l_b \) | 0.247 | m |
| Body Width | \( w_b \) | 0.194 | m |
| Body Height | \( h_b \) | 0.114 | m |
| Thigh and Calf Lengths | \( l_1, l_2 \) | 0.2 | m |
| Wheel Radius | \( R_{wheel} \) | 0.05 | m |

| TABLE II: Joint Actuator Parameters |
|-------------------------------------|
| Parameter | Value | Units |
| Max Torque | 33.5 | Nm |
| Max Joint Speed | 21.0 | Rad/s |

III. OVERVIEW

Having presented the robot model and hardware, in this Section, we highlight the overview of our control architecture and critical parts of our proposed framework. The control architecture and block diagram is present in Fig. 3. We use QP force-based balance controller to maintain balance in various tasks. This balancing control only commands the thigh and calf joint torques. For controlling the wheels, the rolling control enables the robot to maneuver with wheel traction and yaw on command. The details of the balancing control will be explained in Section IV. In order
to control our robot to roll over challenging terrains (e.g., stairs, high obstacles or steep ramps), we also propose a pose optimization framework to solve for optimal configuration of the robot that is collision-free with the terrain while maintaining a good support region for the robot to keep the body balanced. Section V will explain in more details this approach.

IV. FORCE-BASED BALANCE CONTROL

A. Rigid-body Dynamics with Wheels

In this section, we introduce a simplified dynamics model for wheel-legged robots that we can be used effectively in a real-time feedback control for balancing the robot body. Single rigid-body dynamics are commonly used for controlling quadruped robots [24], [15]. However, when the robot high-leverages the wheels in traversing an uneven terrain, it is critical to take into account the impact of wheel traction forces on the robot dynamics. In our work, the simplified dynamics model takes wheel dynamics into consideration. Our control framework has been divided into balance control and rolling control. The balancing controller output is mapped to only the thigh and calf torques, and the wheel torque is controlled by rolling controller. The two decoupled control methods are connected by inputting the wheel dynamics into the simplified rigid body dynamics in QP balancing controller. Considering the wheel dynamics in angular motion while on the ground, shown in Fig. 4a.

\[ I_{\text{wheel}} \cdot \ddot{\omega}_{\text{wheel}} = \tau_{\text{wheel}_i} - F_{t_i} \cdot R_{\text{wheel}}, \]  \hspace{1cm} (1)

where \( I_{\text{wheel}} \) is the rotation inertia of the wheel, \( \ddot{\omega}_{\text{wheel}} \) is the angular acceleration of the wheel, \( \tau_{\text{wheel}_i} \) is the wheel torque of the \( i^{\text{th}} \) leg from control input, and \( F_{t_i} \) is the wheel traction force of the \( i^{\text{th}} \) leg at ground contact point on the rim of the wheel. We choose to neglect the very small rotation inertia value of the wheel dynamics to obtain a linear relation of the wheel torque and traction force, equation (1) is then reduced to:

\[ F_{t_i} = \frac{\tau_{\text{wheel}_i}}{R_{\text{wheel}}}. \]  \hspace{1cm} (2)

Figure 4b illustrates definitions of parameters needed for our rigid body dynamics with wheels. With the addition of wheel traction forces, the simplified dynamics of the robot can be written as:

\[ A_c \cdot F + A_{\text{wheel}} \cdot F_{\text{wheel}} = b, \]  \hspace{1cm} (3)

where

\[ A_c = \begin{bmatrix} I_2 & \cdots & I_2 \\ r_{c_1} \times & \cdots & r_{c_4} \times \end{bmatrix} \]  \hspace{1cm} (4)

\[ A_{\text{wheel}} = \begin{bmatrix} I_2 & \cdots & I_2 \\ r_{\text{wheel}_1} \times & \cdots & r_{\text{wheel}_4} \times \end{bmatrix} \]  \hspace{1cm} (5)

\[ b = \begin{bmatrix} m(\ddot{p}_c + g) \\ I_w \ddot{\omega} \end{bmatrix}. \]  \hspace{1cm} (6)

The term \( F \) is a force vector containing 2D ground reaction forces at the center of each wheel, \( F = [F_{t_1}, x, F_{t_1}, z, \ldots, F_{t_4}, x, F_{t_4}, z]^T \). Similarly, \( F_{\text{wheel}} \) is a column vector containing the wheel traction forces obtained from equation (2), \( F_{\text{wheel}} = [F_{t_1}, 0, \ldots, F_{t_4}, 0]^T \). Note that the wheel only contributes force in the direction of the ground. In equation (4) and (5), \( r_{c_i} \) is the vector of distance from trunk CoM to center of the \( i^{\text{th}} \) wheel and \( r_{\text{wheel}_i} \) is the distance from CoM to ground contact point of the \( i^{\text{th}} \) wheel, \( i = 1, \ldots, 4 \). \( r_{c_i} \times \) is \( [r_{c_i}, z, -r_{c_i}, x] \) and \( r_{\text{wheel}_i} \times = [r_{\text{wheel}_i}, z, -r_{\text{wheel}_i}, x] \). In equation (6), \( \ddot{p}_c \) is the linear acceleration of the robot CoM in 2D (x and z direction), \( g \) is the gravity vector in 2D, \( I_w \) is the rotation inertia of the robot body in the world frame, and \( \ddot{\omega} \) is the angular acceleration of the robot body around y-axis.

B. QP Formulation of Balance Controller

Since the dynamics model [3] presented in the previous section is linear, we can incorporate the dynamics constraints in a quadratic program (QP) as follow. The formulation of this balance controller is inspired by [24] which was developed for quadruped robots using a single rigid body dynamics. In this work, we adopt this principle using our model of rigid body dynamics with wheels for our wheel-legged robots. The balancing control employs a PD control policy of the robot body CoM position. It also makes sure the inequality constraints such as force saturation and friction constraints are stratified in the optimal solution.

In this work, we will design a controller that tends to drive the robot dynamics to the following desired dynamics that
follows a PD control law:
\[ \begin{bmatrix} \ddot{p}_{c,des} \\ \omega_{des} \end{bmatrix} = \begin{bmatrix} K_{np,des} - p_c \\ K_{n,\omega}(\theta_{des} - \theta) \end{bmatrix}. \]  
(7)

The right-hand side of equation (7) contains user input command in terms of desired CoM position, velocity, pitch angle \( \theta_{des} \) and angular velocity. The left-hand side can then be used to represent the desired \( b \) matrix in the dynamics equation (3):
\[ b_{des} = \begin{bmatrix} m(\ddot{p}_{c,des} + g) \\ f_{w} \omega_{des} \end{bmatrix}. \]  
(8)

Then, we can obtain the desired dynamics by driving the left-hand side of the dynamics equation (3) to:
\[ A_c \cdot F \rightarrow b_{des} - A_{wheel} \cdot F_{wheel} \]  
(9)

where the value of \( F_{wheel} \) is dependent on wheel torques, controller by rolling controller in Section IV-C.

Equation (9) can be obtained by the following quadratic program:
\[ F_{opt} = \min_{F \in \mathbb{R}^n} D^T SD + \alpha \| F \|^2 + \beta \| F - F_{opt,prev} \|^2 \]  
(10)

s.t. \( CF \leq d \)  
(11)

where \( D = A_c \cdot F + A_{wheel} \cdot F_{wheel} - b_{des} \). Equation (10) is the cost function of this QP problem, its main goals are driving robot CoM location close to the desired command, minimize the optimal force \( F_{opt} \), and filtering the difference of optimal force at current time step and previous optimal force \( F_{opt,prev} \). These three tasks are weight by \( S \), \( \alpha \), and \( \beta \) to determine the task priorities. Equation (11) summarizes the friction cone constraint and saturation of computed ground reaction force.

The resulting optimal force inputs from the QP problem in equation (10) and (11), \( F_{opt} = [F_{1,z}, F_{1,\omega}, \ldots, F_{4,z}, F_{4,\omega}]^T \) are then mapped to the thigh and calf joint torques for each leg by:
\[ \tau_{QP,i} = J_i^T \begin{bmatrix} F_{1,z} \\ F_{1,\omega} \end{bmatrix}, \]  
(12)

where \( J_i \) is the leg Jacobian matrix of \( i \)th leg.

C. 3D Wheel Rolling Control

While the QP force control provides balance and stability to the wheel-legged robot during motion, the forward velocity and yaw control of the robot can be realized by leveraging the rolling motion of the wheels. With a given CoM velocity command, the wheel torque is calculated using the following feedback law:
\[ \tau_{wheel} = K_{d,\omega}(\dot{\psi}_{wheel,des} - \dot{\psi}_{wheel}), \]  
(13)

where \( \dot{\psi}_{wheel} \) is the measurement of the wheel joint angular velocity, and
\[ \dot{\psi}_{wheel,des} = \ddot{p}_{w,des} \frac{R_{wheel}}{R_{wheel}}, \]  
(14)

with \( \ddot{p}_{w,des} \) being the desired forward velocity. On top of this rolling control based on the input linear velocity command, the controller can also track a desired yaw speed command during rolling motion. This is achieved by assigning a difference \( \Delta \dot{\psi}_{wheel,des} \) in commanded angular speed to the left and right wheel joints, to achieve a feedback turning control. And \( \Delta \dot{\psi}_{wheel} \) is adjusted by a yaw-speed (\( \dot{\psi} \)) controller,
\[ \Delta \dot{\psi}_{wheel} = K_{d,\dot{\psi}}(\dot{\psi}_{des} - \dot{\psi}). \]  
(15)

The combination of QP force-based balance control and rolling control allows the wheel-legged robot to have stable dynamic locomotion over uneven terrain by taking the advantage of wheel rolling traction.

V. POSE OPTIMIZATION

In the previous sections, we have explained the architecture of the hybrid control method that enables stable locomotion of wheel-legged robots only leveraging the wheel rolling motion. However, with only balancing control and wheel rolling control, the robot is unable to pass more complex terrains such as terrain with very steep slope and staircases. We can further manipulate and control the robot to achieve certain poses in order to maintain stability and collision avoidance to overcome challenging terrains indicated above. These poses are terrain-and-task-dependent and are difficult to solve with the current hybrid control framework which relies heavily on a linearized dynamics. Hence, we propose a pose optimization method based on robot kinematics and can solve for optimal poses for a given slope-climbing or multiple-stair-climbing task while ensuring collision-free constraints with the terrain.

A. Problem Formulation

In order to command the wheel-legged robot to climb up a high stair with certain slope angle by leveraging only the rolling wheels, the nominal standing pose on the flat ground has the shortcoming of small stability region and CoM location may fall outside of the support region of the four legs. Hence, in these tasks, we can manipulate the robot to maintain and transform between different poses in order to create a large stability region while the robot is on a slope and while the body is at an significant pitch angle.

In order to decrease the problem computation expense, the pose optimization only need to compute two optimal poses at certain positions in a single-stair climbing task. As Fig. 5 has shown, the two pose locations has the largest kinematic changes during this transition from ground to upper surface. To ensure these critical poses are collision free, we simplified the contact model of the robot to 25 possible contact points placed across the robot trunk and limbs. The trunk, thigh and calf links each has 5 contact points. The total number of the points are determined by trial-and-error from simulation results and is the middle ground of a well-constrained problem vs. computation time.

The optimization method also has great potential of extending its usage to more complex terrains such as multiple stair climbing, by only adding few more poses to solve at more critical locations.
B. Cost Function

The objective of the pose optimization is to find the optimal pose at pose location \( p_{c}^{ref} = [x_{c}^{ref}, z_{c}^{ref}]^T \), the reference pose location is resulted from the terrain information. Hence, the cost function aims to find the closest possible location that satisfy the given NLP constraints,

\[
J = (p_{c}^{ref} - p_{c})^T Q (p_{c}^{ref} - p_{c})
\]  

(16)

\( Q \) is a diagonal weighting matrix.

In this optimization framework, we use kinematic constraints, therefore, a feasible pose solution \( q^* \) should contain only joint angles \( q^* = [q_1, q_2, q_3, q_4]^T \). It is necessary to allow CoM z-direction location delta \( z_{c}^{ref} - z_{c} \) to have certain flexibility in order to solve for the most optimal poses, in another word, we choose the weight in the z-direction location delta to be much smaller than that of x-direction location delta.

C. Constraints

The following constraints are applied in the pose optimization NLP problem:

- Forward kinematics for foot height
- Forward kinematics for foot support region
- Forward kinematics for all contact points
- Joint angle limit: \( q_{min} < q < q_{max} \)

The foot location of our wheel-legged robot is defined as the ground contact point on a wheel. The z-direction locations of each foot in Cartesian coordinate is constraint to be equal to the ground height, to ensure the pose is non-aerial. The x-direction locations of each feet are also constrained to form a sizeable support region for the trunk. Hence, the x-direction location of the rear foot is set to slighted less than the corresponding hip x-direction location. All contact point locations are constrained to avoid collision or contact of the stair terrain.

D. Tracking Controller

Since the optimization outputs only optimal joint and pitch angles, a tracking controller is needed to enable the robot to perform certain pose at desired location and timing. The optimal poses \( q^* \) are tracked by a joint PD controller, while the optimal pitch angle \( \theta^* \) is tracked by QP balancing control. The timing of each pose \( t_1 \) and \( t_2 \) is estimated by the current average wheel joint speed \( \bar{q}_{wheel} \) and terrain stair slope angle \( \gamma \) and height \( h \),

\[
\dot{t}_1 = t + \frac{(\Delta x - R_{\text{wheel}})}{R_{\text{wheel}} q_{\text{wheel}}}
\]  

(17)

\[
\dot{t}_2 = \dot{t}_1 + \frac{(h \sin(\gamma)) + R_{\text{wheel}}}{R_{\text{wheel}} q_{\text{wheel}}}
\]  

(18)

\( t \) is the current timing at the start of estimation.

The joint angle trajectory is linearly interpolated from initial pose to optimal pose 1, from optimal pose 1 to optimal pose 2, and from optimal pose 2 to the final pose (pose sequence shown in Fig. 5).

\[
\tau = \tau_{\text{QP}} + K_{p,q}(q^* - q) + K_{d,q}(\dot{q}^* - \dot{q})
\]  

(19)

The tracking controller works alongside with QP force-based balance control to balance the robot during stair climbing tasks. The resulting control input \( \tau \) in terms of joint torques for thigh and calf joints is a combination of torque from balance controller \( \tau_{\text{QP}} \) and tracking controller, as shown in equation [19].

E. Balance Controller Dynamics on Terrain with Slope

Finally, it is necessary to modify the simplified dynamics for the wheel legged robot model in the QP force controller with the consideration of terrain slope angles, because with aggressive slope angles, the QP balance controller fails to take consideration into the change in coordinate frames of ground reaction forces as well as friction constraints. When all the wheel(s) of the robot are on the slope with an positive angle \( \gamma \), notice that the ground reaction force in z-direction has to be normal to the terrain while the ground reaction force in the x-direction has to be parallel to the terrain. Hence, in the dynamics equation [3], the component \( A_c \) and \( A_{\text{wheel}} \) are modified as follows,

\[
A_{c,\gamma} = \begin{bmatrix} R_\gamma \times & \cdots & R_\gamma \times \\
R_\gamma \times & \cdots & R_\gamma \times 
\end{bmatrix}
\]  

(20)

\[
A_{\text{wheel,}\gamma} = \begin{bmatrix} R_\gamma \times & \cdots & R_\gamma \times \\
R_\gamma \times & \cdots & R_\gamma \times 
\end{bmatrix}
\]  

(21)

where \( R_\gamma \) is 2D rotation matrix by the slope angle \( \gamma \),

\[
R_\gamma = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) \\
-\sin(\gamma) & \cos(\gamma) 
\end{bmatrix}
\]  

(22)

The above modified dynamics formulation will be applied to the tasks in which the terrain has steep slopes.

VI. RESULTS

A. Simulation Results

We validate our control strategy in a high-fidelity simulation in MATLAB and Simulink, with Spatial v2 environment [25]. In simulation, we have successfully commanded the robot to go up to a 0.34m high stair with a slope angle of 80° with the hybrid control of QP balancing and joint PD pose-tracking. Nominal quadruped robot gait cannot climb up such a high stair due to it is constrained by nominal height of the robot. However, in this framework, we can manipulate the robot poses by kinematic optimization and leverage the
wheel traction to overcome such obstacle height. Fig. 6a is a series of snapshots of this simulation and Fig. 7 shows the pitch angle and joint tracking performance during the entire motion.

We have also proved the pose optimization’s capability of extension and versatility to more complex terrain problems by a set of 3-stair climbing simulations with varied stair runs and rises, shown in Fig. 6b and Fig. 6c.

B. Hardware Implementation

We have also successfully validated our proposed approach on the real robot hardware. In hardware experiments, our method with pose optimization has also showed its advantages compared to without pose optimization. Fig. 8 shows snapshots of experiment results comparing the performance of these two approaches. The robot is commanded to roll up a 0.25m high stair with a ramp slope of 30°. It is observed that, using the nominal quadruped pose (without pose optimization), the CoM location falls towards the back of the leg support region and causes significant pitch angle error. In addition, the leg configuration does not favor traversing over slopes due to collision of rear calf joints and the ground. Whereas, with our proposed approach using pose optimization, the CoM location stays centered at support region and the body pitch maintains within 20°. The pitch and joint angle tracking plots of the successful experiment with pose optimization are shown in Fig. 9. This also demonstrates the effectiveness of our balancing control presented in this paper.

Fig. 6: Simulation Snapshots.

We have also successfully validated our proposed approach on the real robot hardware. In hardware experiments, our method with pose optimization has also showed its advantages compared to without pose optimization. Fig. 8 shows snapshots of experiment results comparing the performance of these two approaches. The robot is commanded to roll up a 0.25m high stair with a ramp slope of 30°. It is observed that, using the nominal quadruped pose (without pose optimization), the CoM location falls towards the back of the leg support region and causes significant pitch angle error. In addition, the leg configuration does not favor traversing over slopes due to collision of rear calf joints and the ground. Whereas, with our proposed approach using pose optimization, the CoM location stays centered at support region and the body pitch maintains within 20°. The pitch and joint angle tracking plots of the successful experiment with pose optimization are shown in Fig. 9. This also demonstrates the effectiveness of our balancing control presented in this paper.

Fig. 6: Simulation Snapshots.

We have also successfully validated our proposed approach on the real robot hardware. In hardware experiments, our method with pose optimization has also showed its advantages compared to without pose optimization. Fig. 8 shows snapshots of experiment results comparing the performance of these two approaches. The robot is commanded to roll up a 0.25m high stair with a ramp slope of 30°. It is observed that, using the nominal quadruped pose (without pose optimization), the CoM location falls towards the back of the leg support region and causes significant pitch angle error. In addition, the leg configuration does not favor traversing over slopes due to collision of rear calf joints and the ground. Whereas, with our proposed approach using pose optimization, the CoM location stays centered at support region and the body pitch maintains within 20°. The pitch and joint angle tracking plots of the successful experiment with pose optimization are shown in Fig. 9. This also demonstrates the effectiveness of our balancing control presented in this paper.

Fig. 7: Pose Tracking in Simulation. a) Optimal pitch angle tracking with QP balancing control, and, b) Joint angle tracking, in stair climbing task with height of 0.34m and slope angle of 80°.

We have also successfully validated our proposed approach on the real robot hardware. In hardware experiments, our method with pose optimization has also showed its advantages compared to without pose optimization. Fig. 8 shows snapshots of experiment results comparing the performance of these two approaches. The robot is commanded to roll up a 0.25m high stair with a ramp slope of 30°. It is observed that, using the nominal quadruped pose (without pose optimization), the CoM location falls towards the back of the leg support region and causes significant pitch angle error. In addition, the leg configuration does not favor traversing over slopes due to collision of rear calf joints and the ground. Whereas, with our proposed approach using pose optimization, the CoM location stays centered at support region and the body pitch maintains within 20°. The pitch and joint angle tracking plots of the successful experiment with pose optimization are shown in Fig. 9. This also demonstrates the effectiveness of our balancing control presented in this paper.

Fig. 8: Pose Optimization in Hardware. Experiment snapshots in stair climbing task with height of 0.25m and slope angle of 30°, with and without pose optimization.

We have also successfully validated our proposed approach on the real robot hardware. In hardware experiments, our method with pose optimization has also showed its advantages compared to without pose optimization. Fig. 8 shows snapshots of experiment results comparing the performance of these two approaches. The robot is commanded to roll up a 0.25m high stair with a ramp slope of 30°. It is observed that, using the nominal quadruped pose (without pose optimization), the CoM location falls towards the back of the leg support region and causes significant pitch angle error. In addition, the leg configuration does not favor traversing over slopes due to collision of rear calf joints and the ground. Whereas, with our proposed approach using pose optimization, the CoM location stays centered at support region and the body pitch maintains within 20°. The pitch and joint angle tracking plots of the successful experiment with pose optimization are shown in Fig. 9. This also demonstrates the effectiveness of our balancing control presented in this paper.

Fig. 7: Pose Tracking in Simulation. a) Optimal pitch angle tracking with QP balancing control, and, b) Joint angle tracking, in stair climbing task with height of 0.34m and slope angle of 80°.

We have also successfully validated our proposed approach on the real robot hardware. In hardware experiments, our method with pose optimization has also showed its advantages compared to without pose optimization. Fig. 8 shows snapshots of experiment results comparing the performance of these two approaches. The robot is commanded to roll up a 0.25m high stair with a ramp slope of 30°. It is observed that, using the nominal quadruped pose (without pose optimization), the CoM location falls towards the back of the leg support region and causes significant pitch angle error. In addition, the leg configuration does not favor traversing over slopes due to collision of rear calf joints and the ground. Whereas, with our proposed approach using pose optimization, the CoM location stays centered at support region and the body pitch maintains within 20°. The pitch and joint angle tracking plots of the successful experiment with pose optimization are shown in Fig. 9. This also demonstrates the effectiveness of our balancing control presented in this paper.

VII. CONCLUSIONS

In conclusion, we proposed an effective approach of balancing the 12 DoF wheel-legged robot with QP force-based control that employs a modified simplified dynamics that considers the effects of wheel dynamics. By leveraging the wheel traction in uneven terrain locomotion, we have also proposed an optimization method that solves for favorable poses during stair-climbing tasks. The optimal poses are tracked by a joint PD tracking controller, along with QP balancing controller, in simulation, the robot is able to climb up on a 0.34 m stair with a slope angle of 80° with wheel rolling traction. In hardware implementation, the robot is capable of climbing up on a 0.25 m stair with a slope angle of 30° and proved to have superior performance as compared to normal quadruped poses during such tasks.
REFERENCES

[1] S. Kajita and B. Espiau, “Legged robot,” 2008.
[2] Q. Nguyen, A. Agrawal, X. Da, W. C. Martin, H. Geyer, J. W. Grizzle, and K. Sreenath, “Dynamic walking on randomly-varying discrete terrain with one-step preview,” in *Robotics: Science and Systems*, vol. 2, no. 3, 2017.
[3] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, “Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway,” in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4559–4566.
[4] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, “Mit cheetah 3: Design and control of a robust, dynamic quadruped robot,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.
[5] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, et al., “Anymal-a highly mobile and dynamic quadrupedal robot,” in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 38–44.
[6] M. Raibert, K. Blankespoor, G. Nelson, R. Playter et al., “Bigdog, the rough-terrain quadruped robot,” in *Proceedings of the 17th IFAC World Congress*, 2008, pp. 10823–10825.
[7] A. Bouman, M. F. Ginting, N. Alatur, M. Palieri, D. D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick et al., “Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2518–2525.
[8] S. Seok, A. Wang, M. Y. Chuah, D. J. Hyun, J. Lee, D. M. Otten, J. H. Lang, and S. Kim, “Design principles for energy-efficient legged locomotion and implementation on the mit cheetah robot,” *Ieee/asmese transactions on mechatronics*, vol. 20, no. 3, pp. 1117–1129, 2014.
[9] D. Pongas, M. Mistry, and S. Schaal, “A robust quadruped walking gait for traversing rough terrain,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 1474–1479.
[10] M. Schwarz, T. Rodehutskors, M. Schreiber, and S. Behnke, “Hybrid driving-stepping locomotion with the wheeled-legged robot momaro,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5589–5595.
[11] C. Grand, F. BenAmar, F. Plumet, and P. Bidaud, “Stability and traction optimization of a reconfigurable wheel-legged robot,” *The International Journal of Robotics Research*, vol. 23, no. 10-11, pp. 1041–1058, 2004.
[12] C. Grand, F. BenAmar, F. Plumet, and P. Bidaud, “Decoupled control of posture and trajectory of the hybrid wheel-legged robot hylos,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04*, 2004, vol. 5. IEEE, 2004, pp. 5111–5116.
[13] M. Bjelonic, R. Grandia, O. Harley, C. Galliardi, S. Zimmermann, and M. Hutter, “Whole-body mpc and online gait sequence generation for wheeled-legged robots,” *arXiv preprint arXiv:2010.06322*, 2020.
[14] V. Klemm, A. Morra, L. Galich, D. Manohart, D. Rohr, M. Kamel, Y. de Viragh, and R. Siegwart, “Lqr-assisted whole-body control of a wheeled bipedal robot with kinematic loops,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3745–3752, 2020.
[15] Q. Nguyen, M. J. Powell, B. Katz, J. Di Carlo, and S. Kim, “Optimized jumping on the mit cheetah 3 robot,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7448–7454.
[16] J. Li and Q. Nguyen, “Force-and-moment-based model predictive control for achieving highly dynamic locomotion on bipedal robots,” *arXiv preprint arXiv:2104.00065*, 2021.
[17] B. Katz, J. Di Carlo, and S. Kim, “Mini cheetah: A platform for pushing the limits of dynamic quadrupedal control,” in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6295–6301.
[18] G. Bellisard and K. Byl, “Trajectory optimization for a wheel-legged system for dynamic maneuvers that allow for wheel slip,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 7776–7781.
[19] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, “Gait and trajectory optimization for legged systems through phase-based end-effector parameterization,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.
[20] Y. Luo, Q. Li, and Z. Liu, “Design and optimization of wheel-legged robot: Rolling-wolf,” *Chinese journal of mechanical engineering*, vol. 27, no. 6, pp. 1133–1142, 2014.
[21] G. Besserer, C. Grand, F. B. Amar, F. Plumet, and P. Bidaud, “Locomotion modes of an hybrid wheel-legged robot,” in *Climbing and Walking Robots*. Springer, 2005, pp. 825–833.
[22] F. B. Amar, C. Grand, G. Besserer, and F. Plumet, “Performance evaluation of locomotion modes of an hybrid wheel-legged robot for self-adaptation to ground conditions,” in *ASTRA’04, 8th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, 2004.
[23] L. Ni, F. Ma, and L. Wu, “Posture control of a four-wheel-legged robot with a suspension system,” *IEEE Access*, vol. 8, pp. 152 790–152 804, 2020.
[24] M. Focchi, A. Del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, “High-slope terrain locomotion for torque-controlled quadruped robots,” *Autonomous Robots*, vol. 41, no. 1, pp. 259–272, 2017.
[25] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.