Keep Rollin’ – Whole-Body Motion Control and Planning for Wheeled Quadrupedal Robots

Marko Bjelonic, C. Dario Bellicoso, Yvain de Viragh, Dhionis Sako, F. Dante Tresoldi, Fabian Jenelten and Marco Hutter

Abstract—We show dynamic locomotion strategies for wheeled quadrupedal robots, which combine the advantages of both walking and driving. The developed optimization framework tightly integrates the additional degrees of freedom introduced by the wheels. Our approach relies on a zero-moment point based motion optimization which continuously updates reference trajectories. The reference motions are tracked by a hierarchical whole-body controller which computes optimal generalized accelerations and contact forces by solving a sequence of prioritized tasks including the nonholonomic rolling constraints. Our approach has been tested on ANYmal, a quadrupedal robot that is fully torque-controlled including the non-steerable wheels attached to its legs. We conducted experiments on flat and inclined terrains as well as over steps, whereby we show that integrating the wheels into the motion control and planning framework results in intuitive motion trajectories, which enable more robust and dynamic locomotion compared to other wheeled-legged robots. Moreover, with a speed of 4 m/s and a reduction of the cost of transport by 83 % we prove the superiority of wheeled-legged robots compared to their legged counterparts.

Index Terms—Legged Robots, Wheeled Robots, Motion Control, Motion and Path Planning, Optimization and Optimal Control

I. INTRODUCTION

HEELS are one of the major technological advances of humankind. In daily life, they enable us to move faster and more efficiently as compared to legged-based locomotion. The latter, however, is more versatile and offers the possibility to negotiate challenging environments, which is why combining both strategies into one system, would achieve the best of both worlds.

While most of the advances towards autonomous mobile robots either focus on pure walking or driving, this paper shows how to plan and control trajectories for wheeled-legged robots as depicted in Fig. 1 to achieve dynamic locomotion. We believe that such kinds of systems offer the solution for many robotic tasks as described in [1], e.g., rapid exploration, payload delivery, search and rescue, and industrial inspection.

Fig. 1. The fully torque-controlled quadrupedal robot ANYmal [2] is equipped with four non-steerable, torque-controlled wheels. Thus, the number of actuated joint coordinates \( n_\tau \) and the number of joints \( n_j \) are both equal to 16. A video demonstrating the results can be found at [https://youtu.be/nGLUsyx9Vvc](https://youtu.be/nGLUsyx9Vvc).

A. Related Work

Recent years have shown an active research area focusing on the combination of wheeled and legged locomotion. Most wheeled-legged robots, such as [3]–[8], behave like an active suspension system while driving and do not use their legs as a locomotion alternative to the wheels. While these wheeled-legged robots are using a kinematic approach to generate velocity commands for the wheels, there has been some promising research incorporating the whole-body dynamics of the robot to generate torque commands for each of the joints, including the wheels.

The authors in [9] show a prioritized whole-body compliant control framework that generates motor torques for the upper body of a humanoid robot attached to a wheeled base. The equations of motion, including the nonholonomic constraints, are also incorporated into the control structure of a two-wheeled mobile robot [10]. Justin [11], a wheeled humanoid robot, creates torque commands for each of the wheels using an admittance-based velocity controller. Each of these wheeled platforms, however, is not able to step due to the missing legs, and as such, the robots are only performing wheeled locomotion.

In contrast, DRC-HUBO+ [12] is a wheeled humanoid robot which is able to switch between a walking and a driving configuration. While driving, the robot is in a crouched position, and as such, the legs are not used for locomotion or balancing.
Momaro [13], on the other hand, shows driving and stepping without changing its configuration. This wheeled quadrupedal robot uses a kinematic approach to drive and to overcome obstacles like stairs and steps. Recently, the Centauro robot [14–16] showed similar results over stepping stones, steps and first attempts to overcome stairs, while performing only slow static maneuvers.

There is a clear research gap for wheeled-legged robots. Most of the robots using actuated wheels are not taking into account the dynamic model of the whole-body including the wheels. The lack of these model properties hinders these robots from performing dynamic locomotion during walking and driving. In particular, a wheeled-legged robot produces reaction forces between its wheels and the terrain to generate its motion. The switching of the legs’ contact state, the additional degrees of freedom (DOF) along the rolling direction of each wheel, and the reaction forces, all need to be accounted for in order to reveal the potential of wheeled-legged robots compared to traditional legged systems. In addition, torque control for the wheels is only explored for some slowly moving wheeled mobile platforms. Without force or torque control, the friction constraints related to the no-slip condition cannot be fulfilled, and locomotion is not robust against unknown terrain irregularities. Research areas in traditional legged locomotion [2], [17]–[22], however, offer solutions to bridge these gaps. To this end, the work in [23] shows a generic approach to generate motions for wheeled-legged robots. Due to the formulation of the nonlinear programming (NLP) problem, the computation is too slow to execute in a receding horizon fashion, which is needed for robust execution under external disturbances. Moreover, the same authors verified their NLP algorithm on rather small robots.

So far, Boston Dynamics’ wheeled bipedal robot Handle [24] is the only solution that demonstrated dynamic motions to overcome high obstacles while showing adaptability against terrain irregularities. Due to the missing publications on Handle, there is no knowledge about Boston Dynamics’ locomotion framework.

B. Contribution

This paper shows dynamic locomotion for wheeled quadrupedal robots which combine the mobility of legs with the efficiency of driving. Our main contribution is a whole-body motion control and planning framework which takes into account the additional degrees of freedom introduced by torque-controlled wheels. The motion planner relies on an online zero-moment point (ZMP) [25] based optimization which continuously updates reference trajectories for the free-floating base and the wheels in a receding horizon fashion. These optimized motion plans are tracked by a hierarchical whole-body controller (WBC) which takes into account the nonholonomic constraints introduced by the wheels. In contrast to other wheeled-legged robots, all joints including the wheels are torque controlled. To the best of our knowledge, this work shows for the first time dynamic and hybrid locomotion over flat, inclined and rough terrain for a wheeled quadrupedal robot. Moreover, we show how the same whole-body motion controller and planner are applied to driving and walking without changing any of the principles of dynamics and balance.

II. MODELLED OF WHEELED-LEGGED ROBOTS

We first recall basic definitions of the kinematics and dynamics of robotic systems. Similar to walking robots [17], a wheeled-legged robot is modeled as a free-floating base to which the legs including the wheels as end-effectors are attached. Given a fixed inertial frame $I$ (see Fig. 2), the position from frame $I$ to $B$ with respect to (w.r.t.) frame $I$ and the orientation of frame $B$ w.r.t. frame $I$ are described by $r_{IB} \in \mathbb{R}^3$ and a Hamiltonian unit quaternion $q_{IB}$. The generalized coordinate vector $q$ and the generalized velocity vector $\dot{u}$ are given by

$$q = \begin{bmatrix} r_{IB} \\ q_I \end{bmatrix} \in SE(3) \times \mathbb{R}^{n_j}, \ u = \begin{bmatrix} v_B \\ b\omega_{IB} \\ q_j \end{bmatrix} \in \mathbb{R}^{n_u},$$

where $q_I \in \mathbb{R}^{n_j}$ is the vector of joint coordinates, with $n_j$ the number of joint coordinates, $n_u = 6 + n_j$ is the number of generalized velocity coordinates, $v_B \in \mathbb{R}^3$ is the linear velocity of frame $B$ w.r.t. frame $I$, and $b\omega_{IB} \in \mathbb{R}^3$ is the angular velocity from frame $I$ to $B$ w.r.t. frame $B$. With this convention, the equations of motion for wheeled-legged robots are defined by

$$M(q)\ddot{u} + h(q, u) = S^T \tau + J_S^T \lambda,$$

where $M(q) \in \mathbb{R}^{n_u \times n_u}$ is the mass matrix, $h(q, u) \in \mathbb{R}^{n_u}$ is the vector of Coriolis, centrifugal and gravity terms, $\tau \in \mathbb{R}^{n_\tau}$ is the generalized torque vector acting in the direction of the generalized coordinate vector, with $n_\tau$ the number of actuated joint coordinates, $J_S = [J_{c_1}^T \cdots J_{c_n}^T]^T \in \mathbb{R}^{3n_c \times n_u}$ is the support Jacobian, with $n_c$ the number of limbs in contact, and $\lambda \in \mathbb{R}^{3n_c}$ is the vector of constraint forces. The transpose of the selection matrix $S = [0_{n_\tau \times n_u - n_{z_c}} \mathbb{I}_{n_{z_c} \times n_{z_c}}]$ maps the generalized torque vector $\tau$ to the space of generalized forces.

Fig. 2. The figure illustrates a sketch of the wheeled quadrupedal robot ANYmal and the wheel model used to derive the rolling constraint [3].

Left figure: As discussed in [17], we define a plane frame $P$ which is used as a reference frame in our motion planner. The red and blue arrows visualize the gravito-inertial wrench of the 3D ZMP model described in Section III-C.

Right figure: We differentiate between the leg-fixed and wheel-fixed coordinate frames at the wheel. The leg-fixed wheel frame $W'$ and contact frame $C'$ do not depend on the joint angle $\theta$ of the wheel. In contrast, the wheel-fixed wheel frame $W$ and contact frame $C$ depend on the joint angle $\theta$ of the wheel. Both contact frames are aligned with the local estimation of the terrain normal $n$ and the rolling direction $c_x$ of the wheel.
A. Nonholonomic Rolling Constraint

In contrast to point contacts, the acceleration of the wheel-fixed contact point \( C_i \) of the \( i \)-th leg does not equal zero, i.e., \( \dot{r}_{IC_i} = J_{C_i} \ddot{u} + J_{C_i} u = 0 \). Given the wheel model in Fig. 2 it can be shown that the resulting contact acceleration of a wheel is defined by

\[
\dot{r}_{IC_i} = J_{C_i} \ddot{u} + J_{C_i} u = \begin{bmatrix}
R_{IW_i} & 0 \\
r_0 \psi W_i \cos(\phi W_i)(\chi W_i + \theta) \\
r_0 \chi W_i + \theta (\chi W_i + \theta + \psi W_i \sin(\phi W_i))
\end{bmatrix},
\]

where \( R_{IW_i} \in SO(3) \) represents the rotation matrix that projects the components of a vector from the wheel frame \( W_i \) to the inertial frame \( I \). \( r_0 \) is the wheel radius, and \( \theta_i \) is the joint angle of the wheel. Using an intrinsic \( \mathbf{z} - \vec{x} - \vec{y} \) Euler parameterization, the yaw, roll, and pitch angle of the wheel frame \( W_i \) w.r.t. the inertial frame \( I \) are given by \( \psi W_i, \phi W_i, \) and \( \theta W_i \), respectively.

By setting \( \phi W_i = 0 \) and \( \psi W_i = 0 \), we obtain the acceleration for the planar case, i.e., \( \dot{r}_{IC_i} = R_{IW_i} \dot{x} \). Given the wheel model in Fig. 2, it can be shown that the resulting contact acceleration is aligned with the estimated terrain normal and whose projection in the plan frame is located at the footprint center projected onto the local terrain along the terrain normal.

III. MOTION PLANNING

The dynamic model of a wheeled legged robot \( 2 \) includes significant nonlinearities to be handled by the motion planner. Due to this complexity, the optimization problem becomes prone to local minima and it can be demanding to solve in real-time on-board \( 20 \). To overcome these challenges, our approach breaks down the whole-body planning problem into center of mass (COM) and foothold motion optimization \( 17 \). We simplify the system dynamics to a ZMP model for motion planning of the COM. The reference footholds for each leg are obtained by solving a separate optimization problem.

Fig. 3 gives an overview of the entire whole-body motion control and planning framework. The foothold optimizer, motion optimizer, and WBC modules are solving separate optimization problems in parallel such that there is no interruption between them \( 17 \). We generate all motions w.r.t. the plan frame \( P \) introduced in Section II-B. In the following, we describe each module of the motion planner.

A. Contact Scheduler

The contact schedule defines periodic sequences of lift-off and touch-down events for each leg. Based on a gait pattern library, each gait defines the timings for each leg over a
stride, e.g., the contact scheduler block in Fig. 3 illustrates the gait pattern for a trotting gait. With this formulation, driving is defined by a gait pattern where each leg is scheduled to stay in contact, and no lift-off events are set.

B. Foothold Optimizer

Given a reference in terms of linear velocity \( v_{ref}^B = [v_{ref, x}^B, v_{ref, y}^B, 0]^T \) and angular velocity \( \omega_{ref}^B = [0, 0, \omega_{ref, z}^B]^T \) of the base, and the contact schedule, desired footholds are generated for each leg. Based on the contact schedule and footholds, a sequence of support polygons are generated, where each polygon is defined by the convex hull of expected footolds, e.g., the green polygon in Fig. 2 as well as its time duration in seconds.

While walking, we formulate a quadratic programming (QP) problem which optimizes over the \( x \) and \( y \) coordinates of each foothold [17]. Costs, which are added to the QP problem, penalize the distance between the optimized foothold locations and different contributions to the computation of the footholds. We assign default foothold positions which define the standing configuration of the robot. Footholds are projected using the high-level reference velocity and assuming constant reference velocity throughout the optimization horizon. To ensure smoothness of the footholds, we penalize the deviation from previously computed footholds. Finally, we rely on an inverted pendulum model to stabilize the robot’s motion [19]. Inequality constraints are added to avoid collisions of the feet and to respect the maximum kinematic extension of each leg. Given the previous stance foot position and the optimized footholds, a sequence of support polygons are generated, based on the contact schedule, penalize the distance between the optimized foothold and the green polygon in Fig. 2, as well as its time duration.

Given the previous stance foot position and the optimized footholds, a sequence of support polygons are generated, based on the contact schedule, penalize the distance between the optimized foothold and the green polygon in Fig. 2, as well as its time duration.

C. Motion Optimizer

The motion optimizer generates operational space references for the \( x, y \) and \( z \) coordinates of the whole-body COM given the support polygon sequence and the robot state [17]. The resulting nonlinear optimization framework is described in the following sections.

1) Motion plan parameterization: The \( x, y \) and \( z \) coordinates of the COM trajectory are parametrized as a sequence of quintic splines [17], i.e., the position, velocity and accelerations of the COM are given by \( p_{COM}(t) = T(t)\alpha_k \in \mathbb{R}^3 \), \( \dot{p}_{COM} = T(t)\dot{\alpha}_k \in \mathbb{R}^3 \), and \( \ddot{p}_{COM} = T(t)\ddot{\alpha}_k \in \mathbb{R}^3 \), with \( T(t) = \text{diag}(\eta(t), \eta(t), \eta(t)) \in \mathbb{R}^{3 \times 3}, \eta(t) = [t^5 \ t^4 \ t^3 \ t^2 \ t \ 1], \ t \in [0, \bar{t} + \Delta t_k], \) where \( \bar{t} \) is the sum of time durations of all previous splines, and \( \Delta t_k \) is the time duration of the \( k \)-th spline. All coefficients of spline \( i \) are stored in \( \alpha_k = [\alpha_k^x, \alpha_k^y, \alpha_k^z]^T \in \mathbb{R}^{18} \). Finally, we solve for the vector of optimization parameters which is obtained by stacking together all spline coefficient vectors \( \alpha_k \).

2) Optimization problem: The motion optimization problem is expressed as a nonlinear optimization problem with objective \( f(\xi) \), equality constraints \( c(\xi) \), and inequality constraints \( h(\xi) \). The problem is described by

\[
\begin{align*}
\text{minimize} & \quad f(\xi) \\
\text{subject to} & \quad c(\xi) = 0, \quad h(\xi) \geq 0,
\end{align*}
\]

where \( \xi \) is the vector of optimization variables given in Section III-C1, i.e., optimal spline coefficients are computed. A sequential quadratic programming (SQP) algorithm [27] is used to solve (4) continuously over a time horizon of \( \tau \) seconds. Table I summarizes each objective and constraint used in this work.

3) ZMP inequality constraint: To ensure dynamic stability of the planned motions, an inequality constraint on the ZMP position \( p_{ZMP} \in \mathbb{R}^3 \) is included in the motion optimization, where \( p_{zmp} = n \times m_{Oi}^g (\eta(t) p_{f_i}^g) \) [28]. Here, \( m_{Oi}^g \in \mathbb{R}^3 \) and \( p_{f_i}^g \in \mathbb{R}^3 \) are the components of the gravito-inertial wrench [29], with \( m_{Oi}^g = m \cdot p_{COM} \times (g - \bar{p}_{COM}) - \bar{l} \) and \( p_{f_i}^g = m \cdot (g - \bar{p}_{COM}) \), where \( m \) is the mass of the robot, \( \bar{l} \in \mathbb{R}^3 \) is the angular momentum of the COM, and \( g \in \mathbb{R}^3 \) is the gravity vector. Fig. 2 shows a sketch of the gravito-inertial wrench acting at the COM. As in [17], we assume that \( l = 0 \).

As illustrated in Fig. 2, the ZMP position \( p_{ZMP} \) is constrained to lie inside the support polygon. This stability criterion is formulated as a nonlinear inequality constraint given by [17]

\[
[p \ q \ 0] p_{ZMP} + r^T \geq 0,
\]

Table I contains the motion optimization problem described in [17].

<table>
<thead>
<tr>
<th>Type</th>
<th>Task</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective</td>
<td>Minimize COM acceleration</td>
<td>Smooth motions</td>
</tr>
<tr>
<td>Objective</td>
<td>Minimize deviation to previous solution ( \xi_{prev} )</td>
<td>Smooth motions</td>
</tr>
<tr>
<td>Objective</td>
<td>Track a high-level reference trajectory ( \pi ) (path regularizer) ( \forall \xi )</td>
<td>Reference tracking</td>
</tr>
<tr>
<td>Soft constraint (lin.-quad.)</td>
<td>Minimize deviation to initial &amp; final conditions ( \forall \xi )</td>
<td>Disturbance rejection &amp; tracking</td>
</tr>
<tr>
<td>Soft constraint (lin.-quad.)</td>
<td>Limit overshoots ( \forall \xi )</td>
<td>Avoid kinematic limits of legs</td>
</tr>
<tr>
<td>Constraint (lin. eq.)</td>
<td>Junction constraints ( \forall ) pairs of adjacent splines ( k, k + 1 \neq \xi )</td>
<td>Continuity</td>
</tr>
<tr>
<td>Constraint (lin. ineq.)</td>
<td>Push Contact Constraints</td>
<td>Legs can only push the ground</td>
</tr>
<tr>
<td>Constraint (nonlin. ineq.)</td>
<td>ZMP criterion</td>
<td>Stability</td>
</tr>
<tr>
<td>Soft constraint (nonlin.)</td>
<td>Soften initial ZMP constraints</td>
<td>Relaxation</td>
</tr>
</tbody>
</table>

\( ^5 \) A foothold is the contact position \( C \) of a grounded leg.
where \( d = [p \ q \ r]^T \) are the coefficients of the line that goes through the edge of a support polygon.

4) Deformation of support polygons while driving: In contrast to point feet, the contact locations, and therefore footholds, are not stationary while driving. The support polygon sequence which is needed to fulfill the inequality constraint in (5) is deformed over time. For this purpose, we assume that the number of edges stays constant and therefore, one spline is sufficient to describe the motion of the COM.

First, the expected foothold position for the optimization horizon \( \tau \) is computed as a function of the reference velocities \( v_B^\text{ref} \) and \( \Omega_B^\text{ref} \). The reference velocities are assumed to be constant over the optimization horizon. Using the time-integrated Rodrigues’s formula, the expected foothold position \( p_{\tau,i} \in \mathbb{R}^3 \) of leg \( i \) is computed by

\[
p_{\tau,i} = p_{0,i} + R(\tau \Omega_B^\text{ref}) \\
\frac{1}{\omega_{B,z}^\text{ref}} \begin{bmatrix}
\sin(\omega_{B,z}^\text{ref} \tau) & -1 + \cos(\omega_{B,z}^\text{ref} \tau) & 0 \\
1 - \cos(\omega_{B,z}^\text{ref} \tau) & \sin(\omega_{B,z}^\text{ref} \tau) & 0 \\
0 & 0 & 1
\end{bmatrix} v_B^\text{ref},
\]

where \( p_{0,i} \in \mathbb{R}^3 \) is the current foothold position. If \( \omega_{B,z}^\text{ref} \approx 0 \), the solution becomes \( p_{\tau,i} = p_{0,i} + \tau v_B^\text{ref} \).

Given the coefficients which describe an edge that belongs to the current and expected support polygon, i.e., \( d_0 \in \mathbb{R}^3 \) and \( d_\tau \in \mathbb{R}^3 \), the deformed edge coefficient vector \( d_k(t) \) at time \( t \) is computed by interpolating \( d_0 \) and \( d_\tau \), i.e.,

\[
d(t) = (1 - \frac{t - \hat{t}}{\tau}) d_\tau + \frac{t - \hat{t}}{\tau} d_0.
\]

IV. WHOLE-BODY CONTROLLER

The operational space reference trajectories of the COM and wheels are tracked by a WBC which is based on the hierarchical optimization (HO) framework described in [17], [26]. We compute optimal generalized accelerations \( \dot{u}^* \) and contact forces \( \lambda^* \) which are collected in the vector of optimization variables \( \xi^* = [\dot{u}^* T \lambda^*]^T \in \mathbb{R}^{n_u+3n_c} \), where all symbols are introduced in Section II.

The WBC is formulated as a cascade ofQP problems composed of linear equality and inequality tasks, which are solved in a strict prioritized order [50]. A task \( T_p \) with priority \( p \) is defined by

\[
T_p : \begin{cases}
W_{eq,p}(A_p \xi - b_p) = 0 \\
W_{ineq,p}(D_p \xi - f_p) \leq 0
\end{cases},
\]

where the linear equality constraints are defined by \( A_p \) and \( b_p \), the linear inequality constraints are defined by \( D_p \) and \( f_p \), and the diagonal positive-definite matrices \( W_{eq,p} \) and \( W_{ineq,p} \) weigh tasks on the same priority level.

A. Prioritized Tasks

The highlighted tasks in Table [II] are specifically tailored for wheeled-legged robots, and the following sections describe each of these tasks in more detail. For the remaining tasks, we rely on the same implementation as used for traditional legged robots [26].

<table>
<thead>
<tr>
<th>Priority</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Floating base equations of motion</td>
</tr>
<tr>
<td></td>
<td>Torque limits and friction cone</td>
</tr>
<tr>
<td></td>
<td>Nonholonomic rolling constraint</td>
</tr>
<tr>
<td>2</td>
<td>COM linear and angular motion tracking</td>
</tr>
<tr>
<td></td>
<td>Swing leg motion tracking</td>
</tr>
<tr>
<td></td>
<td>Swing wheel rotation minimization</td>
</tr>
<tr>
<td></td>
<td>Ground leg motion tracking</td>
</tr>
<tr>
<td>3</td>
<td>Contact force minimization</td>
</tr>
</tbody>
</table>

Floating base equations of motion: The optimization vector \( \xi \) is constrained to be consistent with the system dynamics. Torque limits and friction cone: Inequality constraint tasks are added to the optimization problem to avoid that the computed torques exceed the minimum and maximum limit of each actuator. Similar, the contact forces \( \lambda \) need to lie inside the friction cone which is approximated by a friction pyramid and aligned with the normal vector \( n \) of the estimated contact surface shown in Fig. [2].

Nonholonomic rolling constraint: The solution found by the optimization needs to take into account the nonholonomic rolling constraint [3]. This is expressed as an equality constraint given by

\[
[J_S \ 0_{3n_c \times 3n_c}] \xi_d = -J_S \dot{u} + [\dot{r}_{1C_1}^T \ldots \dot{r}_{1C_{n_c}}^T]^T,
\]

where the terms \( \dot{r}_{1C_1} \ldots \dot{r}_{1C_{n_c}} \) on the right side of the equation are the centripetal accelerations of each contact point \( n_c \) derived in [3].

COM linear and angular motion tracking: Similar to the swing leg motion tracking task, the operational space references of the COM are tracked by equality constraint tasks.

Swing leg motion tracking: Given the operational space references of the wheels’ contact points \( p_{1C_1}, \ldots p_{1C_{n_c}} \), the motion tracking task of each swing leg \( i \) is formulated by

\[
[J_{C_i} \ 0_{3n_c \times 3n_c}] \xi_d = R_{f,p}(p_{1C_1}^d - K_p(p_{1C_1}^d - p_{1C_1}^r)) + K_d(p_{1C_1} - p_{1C_1}^d) - J_{C_i} \dot{u},
\]

where \( K_p, K_d \in \mathbb{R}^{3 \times 3} \) are diagonal positive definite matrices which define proportional and derivative gains. Note that all measured values, i.e., \( J_{C_i}, p_{1C_1}, \) and \( p_{1C_1}^d \), are independent of the wheel angle \( \theta \) (as discussed in the footnotes of Section II-A).

Swing wheel rotation minimization: For each swing leg \( i \), the wheel’s rotation is damped by adding the task

\[
[S_{W_i} \ 0_{3n_c \times 3n_c}] \xi_d = -k_d \dot{\theta}_i,
\]

where \( S_{W_i} \in \mathbb{R}^{3n_c \times 3n_c} \) is a matrix which selects the row of \( \xi_d \) containing the wheel of leg \( i \), \( k_d \) is a derivative gain, and \( \dot{\theta}_i \) is the wheel’s rotational speed.

Ground leg motion tracking: To track the desired motion of the grounded legs, we constrain the accelerations in the direction of the rolling direction \( c_x \). Given the operational
space references of the wheels’ contact points \( p_{IC}^d \), \( p_{IC}^r \), \( p_{IC}^d \), \( p_{IC}^r \), the motion tracking task of each ground leg \( i \) is formulated by

\[
\pi_{\text{eq}}(\begin{bmatrix} J_{C_i}^T \ 0_{3n_c \times 3n_c} \end{bmatrix} \xi_d) = \pi_{\text{eq}}(R_{IP}(p_{IC}^d - p_{IC}^d) + K_p(p_{IC}^d - p_{IC}^d)) - J_{c_i}^r u, \]

where \( \pi_{\text{eq}}(a) \) is the projection of a vector \( a \) onto the vector \( c_x \).

Contact force minimization: Finally, the contact forces \( \lambda \) are minimized to reduce slippage.

**B. Torque Generation**

Given the optimal solution \( \xi^* \), the desired actuation torques \( \tau_d \), which are sent to the robot, are computed by

\[
\tau_d = M_j(q)u^* + h_j(q, u) - J_{S_j}^r \lambda^*, \tag{13}
\]

where \( M_j(q) \), \( h_j(q, u) \), and \( J_{S_j} \) are the lower rows of the equations of motion in (4) relative to the actuated joints.

**V. EXPERIMENTAL RESULTS AND DISCUSSION**

To show the benefits and validity of our new approach, this section reports on experiments conducted on a real quadrupedal robot equipped with non-steerable, torque-controlled wheels. The robot is driven using external velocity inputs coming from a joystick. All computation was carried out by the PC (Intel i7-5600U, 2.6 - 3.2GHz, dual-core 64-bit) integrated into the robot. A video showing the results accompanies this paper.

The WBC runs together with state estimation in a 400 Hz loop. A novel state estimation algorithm based on [31] is used to generate an estimation of the robot’s position, velocity, and orientation w.r.t. an inertial coordinate frame. Similar to [32], we fuse data from an inertial measurement unit (IMU) as well as the kinematic measurements from each actuator (including the wheels) to acquire a fast state estimation of the robot. The open-source Rigid Body Dynamics Library [33] (RBDL) is used for modeling and computation of kinematics and dynamics based on the algorithms described in [34]. We use a custom SQP algorithm to solve the nonlinear optimization problem in Section [III-C2] which solves the nonlinear problem by iterating through a sequence of QP problems. Each QP problem is solved using QuadProg++ [35] which uses the Goldfarb-Idnani active-set method [36]. Depending on the gait, the motion optimization in Section [III-C] runs between 100 and 200 Hz.

**A. Indoor Environment: Flat Terrain**

We performed driving and walking in an indoor environment, and the results are illustrated in Fig. [3]. The three-dimensional plot shows the measured trajectories of the front legs, hind legs, and the COM. In addition, the zoomed-in plot depicts the transitions between driving and walking in a corner. As discussed in [37], the robot is able to drive small curvatures although the robot is equipped with non-steerable wheels. By yawing the base of the robot, the wheels are turning w.r.t. an inertial frame. For larger curvatures, the robot needs to step. The results successfully prove the omnidirectional capabilities of the robot.

**B. Indoor Environment: Inclined Terrain**

Fig. [4] depicts the COM motion tracked by the controller while ANYmal is driving blindly over two inclines and

![Fig. 4. The robot ANYmal drives with a speed of 0.7 m/s over two inclines of a height of approximately 30% of ANYmal’s leg length and the red line depicts the COM trajectory. (Available at https://youtu.be/nGLUsyx9Vvc?t=20).](https://youtu.be/nGLUsyx9Vvc)

![Fig. 5. The robot ANYmal is driving and walking in an indoor environment (Available at https://youtu.be/nGLUsyx9Vvc?t=103). The three-dimensional plot shows estimated measurements of the robot where the red, blue and green lines depict the contact trajectories of the front legs, the contact trajectories of the hind legs, and the COM trajectories w.r.t. the inertial frame I. The zoomed-in figure shows transitions between driving and walking while the robot is performing a 90 degrees turn.](https://youtu.be/nGLUsyx9Vvc)
Fig. 6. The plots show the desired motion (i.e., the optimized trajectories of the motion planner) of the COM and the right front leg during the driving maneuver in Fig. 4 (Available at https://youtu.be/nGLUsyx9Vvc?t=20). The executed trajectories are almost identical to the desired motion shown here, and thus, the tracking error is negligible. This is due to the fast update rate (up to 200 Hz) of the motion optimizer and the reinitialization of the optimization problem after every iteration with the measured state of the robot.

Fig. 6 illustrates the optimized trajectories of the motion planner while driving over the inclined terrain. Thanks to torque control, the robot adapts naturally to the unseen terrain irregularities while maintaining the COM height. Moreover, the COM motion is unaffected by the two obstacles although the robot drives at a speed of 0.7 m/s. In addition, none of the wheels violates the friction constraints related to the no-slip condition.

C. Outdoor Environment: Crossing a Street

We conducted an outdoor experiment where we validated the performance of the robot under real-world conditions. Since the robot is able to drive fast and efficiently while being able to overcome obstacles, it applies to real-world tasks such as payload delivery. For this purpose, we conducted an experiment where the robot’s task is to cross a street. As can be seen in Fig. 7, the robot is able to drive down a step and to walk over another one. In addition, the lower left image illustrates how the robot rotates its base around the yaw direction to change its driving direction. This experiment also highlights the significant advantages of wheeled-legged robots compared to traditional walking robots. The robot is able to drive down steps with 1 m/s without the need for terrain perception. Moreover, the lower right image of Fig. 7 shows the robot driving down a stair with 1 m/s without the need to step, confirms the advantage.

D. High Speed and Low Cost of Transport

The computation of the mechanical cost of transport (COT) is based on the work in [37]. On flat terrain, the robot achieves a COT of 0.1 while driving 2 m/s and the mechanical power consumption is 63.64 W. A comparison to traditional walking and skating with passive wheels [37] shows that the COT is lower by 83 % w.r.t. the trotting gait and by 17 % w.r.t. skating motions. In addition, with 4 m/s we broke ANYmal’s maximum speed record of 1.5 m/s given in [38].

VI. CONCLUSIONS

In this work, we show a whole-body motion control and planning framework for a quadrupedal robot equipped with non-steerable, torque-controlled wheels as end-effectors. The mobile platform combines the advantages of legged and wheeled robots. In contrast to other wheeled-legged robots, we show for the first time dynamic motions over flat and inclined terrains as well as over steps. These are enabled thanks to the tight integration of the wheels into the motion planning and control framework. For the motion optimization, we rely on a 3D ZMP approach which updates the motion plan continuously. This motion plan is tracked by a hierarchical WBC which considers the nonholonomic contact constraint introduced by the wheels. Thanks to torque control, the robot does not violate the contact constraints and the fast update rates of the motion control and planning framework make the robot robust in the face of unpredictable terrain.

We aim to demonstrate further the application of the system to real-world tasks by conducting additional outdoor experiments. Future research will focus on hybrid locomotion strategies, i.e., walking and driving at the same time. To this end, promising initial results of a novel trajectory optimization for wheeled-legged quadrupedal robots further expand on the current motion planner presented by optimizing both COM and foot trajectories in a single optimization using linearized ZMP constraints [39]. In addition, perceptive motion planning
over a long time horizon in challenging environments is still an unsolved problem for wheeled-legged and legged robots.

ACKNOWLEDGMENT

The authors would like to thank Vassilios Tsounis for his support during the development of the wheel actuator firmware. Our gratitude goes to Francisco Giraldez Gámez and Christian Gehring who helped with the preliminary investigation of the rolling constraint. Furthermore, we thank Anna Beaurageur for her comments on the final version of the paper.

REFERENCES