Rolling in the Deep – Hybrid Locomotion for Wheeled-Legged Robots using Online Trajectory Optimization

Marko Bjelonic[1], Prajish K. Sankar[2], C. Dario Bellicoso[3], Heike Vallery[4] and Marco Hutter[1]

***Abstract*—Wheeled-legged robots have the potential for highly agile and versatile locomotion. The combination of legs and wheels might be a solution for any real-world application requiring rapid, and long-distance mobility skills on challenging terrain. In this paper, we present an online trajectory optimization framework for wheeled quadrupedal robots capable of executing hybrid walking-driving locomotion strategies. By breaking down the optimization problem into a wheel and base trajectory planning, locomotion planning for high dimensional wheeled-legged robots becomes more tractable, can be solved in real-time on-board in a model predictive control fashion, and becomes robust against unpredicted disturbances. The reference motions are tracked by a hierarchical whole-body controller that sends torque commands to the robot. Our approach is verified on a quadrupedal robot with non-steerable wheels attached to its legs. The robot performs hybrid locomotion with a great variety of gait sequences on rough terrain. Besides, we validated the robotic platform at the Defense Advanced Research Projects Agency (DARPA) Subterranean Challenge, where the robot rapidly mapped, navigated and explored dynamic underground environments.**

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

I. INTRODUCTION

LEGGED robots offer the possibility of negotiating challenging environments and, thus, are versatile platforms for various types of terrains [1]. In research and industry, there is an emphasis on replicating nature to improve the hardware design and algorithmic approach of robotic systems [2], [3]. Even with extensive research, matching the locomotion skills of conventional legged robots to their natural counterparts remains elusive. In contrast, wheels offer a chance to extend some capabilities, particularly speed, of these legged robotic systems beyond those of their natural counterparts, which can be crucial for any task requiring rapid and long-distance mobility skills in challenging environments. With this motivation, the central contribution of this work involves locomotion planning on a wheeled-legged robot to perform dynamic hybrid[1] walking-driving motions on various terrains, as shown in Fig. 1.

Fig. 1. The fully torque-controlled quadrupedal robot ANYmal [4] equipped with four non-steerable, torque-controlled wheels. The robot is traversing over a wooden plank (top images), on rough terrain (left middle image). In addition, the robot rapidly maps, navigates and searches dynamic underground environments at the DARPA Subterranean Challenge (lower images), and the robot's wheels are equipped with chains to traverse the muddy terrain (right middle image). A video can be found at https://youtu.be/ukY0vyM-yfY.

Manuscript received: September, 10, 2019; Revised: December, 18, 2019; Accepted: January, 18, 2020.

This paper was recommended for publication by Editor Nikos Tsagarakis upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by the Swiss National Science Foundation (SNF) through the National Centres of Competence in Research Robotics (NCCR Robotics) and Digital Fabrication (NCCR dfab). Besides, it has been conducted as part of ANYmal Research, a community to advance legged robotics.

Correspondence should be addressed to Marko Bjelonic.

[1] M. Bjelonic and M. Hutter are with the Robotic Systems Lab, ETH Zürich, 8092 Zürich, Switzerland. marko.bjelonic@mavt.ethz.ch

[2] P. K. Sankar is a student at the Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, 2628 CD Delft, Netherlands and was with the Robotic Systems Lab, ETH Zürich, 8092 Zürich, Switzerland at the time of this study.

[3] C. D. Bellicoso is with Boston Dynamics, 02451 Waltham, Massachusetts, United States and was with the Robotic Systems Lab, ETH Zürich, 8092 Zürich, Switzerland at the time of this study.

[4] H. Vallery is with the Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, 2628 CD Delft, Netherlands.

Digital Object Identifier (DOI): see top of this page.

[1] In our work, *hybrid* locomotion denotes simultaneous walking and driving.

A. Related Work

The online generation of optimal solutions for dynamic motions has been an active research area for conventional legged robots. Methods like trajectory optimization (TO) and model predictive control (MPC) are prevalent and recommended in the literature for aiding robots to be reactive against external disturbances and modeling errors. Finding control policies for performing walking motions in an articulated mobile robot is an involved task because of the system's many degrees of freedom (DOF) and its nonlinear dynamics. This demands substantial computational power and introduces the challenge of overcoming local minima, making on-the-fly computations hard.

In the literature concerning wheeled-legged robots, hybrid walking-driving motions are scarce. The focus is mostly on statically-stable driving motions where the legs are used for active suspension alone [5]–[10]. These applications do not show any instance of wheel lift-offs. Hence, sophisticated motion planning for the wheels is unnecessary and, therefore, usually skipped.

Agile motions over steps and stairs are demonstrated for the first time in our previous work [11], where a hierarchical whole-body controller (WBC) tracks the motion trajectories that include the rolling conditions associated with the wheels. The robot can execute walking and driving motions, but not simultaneously due to missing wheel trajectories over a receding horizon. As such, the robot needs to stop and switch to a pure walking mode to overcome obstacles. The work in [12] extends the approach by computing base and wheel trajectories in a single optimization framework. This approach, however, decreases the update rate to 50 Hz, and no hybrid walking-driving motions are shown on the real robot.

CENTAURO, a wheeled-legged quadruped with a humanoid upper-body, performs a walking gait with automatic footstep placement using a linear MPC framework [13]. The authors, however, only perform walking maneuvers without making use of the wheels. In contrast, the path planner in [14] shows driving and walking motions in simulation without considering the robot's dynamics. Among the robots that employ hybrid walking-driving motions, Jet Propulsion Laboratory's (JPL) *Robosimian* uses a TO framework [15], but for passive wheels and results are only shown in a simulation. *Skaterbots* [16] provide a generalized approach to motion planning by solving a nonlinear programming (NLP) problem. This approach, however, is impractical to update online in a receding horizon fashion, i.e., in a MPC fashion, due to excessive computational demand.

Given state of the art, we notice a research gap in trajectory generation methods for hybrid walking-driving motions on legged robots with actuated wheels, which can be both robust on various terrains and be used on-the-fly. Fortunately, research in traditional legged locomotion offers solutions to bridge this gap. The quadrupedal robot *ANYmal* (without wheels) performs highly dynamic motions using MPC [17], [18] and TO [19], [20] approaches. Impressive results are shown by *MIT Cheetah*, which performs blind locomotion over stairs [21] and jumps onto a desk with the height of 0.76 m [22]. The quadrupedal robot *HyQ* shows an online, dynamic foothold adaptation strategy based on visual feedback [23]. Therefore, we conjecture that extending these approaches to wheeled-legged systems can aid in producing robust motions.

B. Contribution

In our work, we present an online TO framework for wheeled-legged robots capable of running in a MPC fashion by breaking the problem down into separate wheel and base TOs. The former takes the rolling constraints of the wheels into account, while the latter accounts for the robot's balance during locomotion using the idea of the zero-moment point (ZMP) [24]. A hierarchical WBC [11] tracks these motions by computing torque commands for all joints. Our *hybrid locomotion framework* extends the capabilities of wheeled-legged robots in the following ways:

1) Our framework is versatile over a wide variety of gaits, such as pure driving, statically stable gaits, dynamically stable gaits, and gaits with full-flight phases.

2) We generate wheel and base trajectories for hybrid walking-driving motions in the order of milliseconds. Thanks to these fast update rates, the resulting motions are robust against unpredicted disturbances, making real-world deployment of the robot feasible. Likewise, we demonstrate the performance of our system at the DARPA Subterranean Challenge, where the robot autonomously maps, navigates and searches dynamic underground environments.

II. MOTION PLANNING

The whole-body motion planner is based on a task synergy approach [25], which decomposes the optimization problem into wheel and base TOs. By breaking down the problem into these two tasks, we hypothesize that the issue of locomotion planning for high-dimensional (wheeled-)legged robots becomes more tractable. The optimization can be solved in real-time in a MPC fashion, and with high update rates, the locomotion can cope with unforeseen disturbances.

The main idea behind our approach is visualized in Fig. 2. Given a fixed gait pattern and the reference velocities[2] with respect to (w.r.t.) the robot's base frame B as shown in Fig. 3, i.e., the linear velocity vector of its center of mass (COM) $\boldsymbol{v}_{\mathrm{ref}}$ and the angular velocity vector $\boldsymbol{\omega}_{\mathrm{ref}} = \begin{bmatrix} 0 & 0 & \omega_{\mathrm{ref}} \end{bmatrix}^T$, desired motion plans are generated in two steps, where the wheel TO is followed by a base TO which satisfies the ZMP [24] stability criterion. The latter simplifies the system dynamics for motion planning of the COM to enable real-time computations onboard. Finally, a controller tracks these motion plans by generating torque commands which are sent to the robot's motor drives. Due to this decomposition of the locomotion problem, the wheel TO, the base TO, and the tracking controller can run in parallel.

The following two sections discuss the main contribution of our work and show how the locomotion of the independent wheel and base TOs are synchronized to generate feasible motion plans.

[2]The reference velocities are generated from an external source, e.g., an operator device, or a navigation planner.

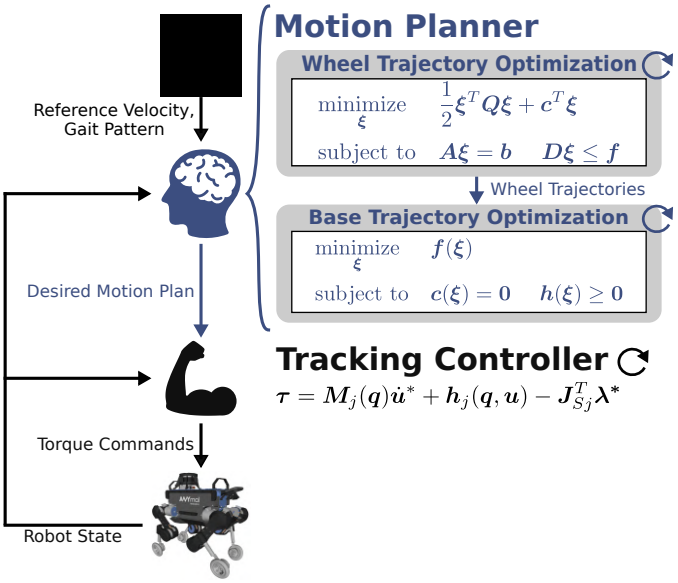

Fig. 2. Overview of the motion planning and control structure. The motion planner is based on a ZMP approach, which takes into account the optimized wheel trajectories and the state of the robot. The hierarchical WBC, which optimizes the whole-body accelerations $\dot{\boldsymbol{u}}^*$ and contact forces $\boldsymbol{\lambda}^*$, tracks the operational space references. Finally, torque references $\boldsymbol{\tau}$ are sent to the robot. The wheel TO, base TO, and WBC can be parallelized due to the hierarchical structure.

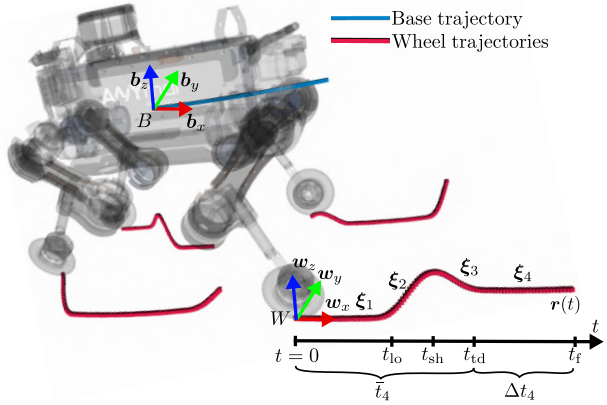

Fig. 3. Timings and coordinate frames. The figure shows a sketch of the wheel and base trajectory. The wheel trajectories are optimized for each of the wheels separately and w.r.t. the coordinate frame W whose z-axis is aligned with the estimated terrain normal, and whose x-axis is perpendicular to the estimated terrain normal and aligned with the rolling direction of the wheel. The origin of W is at the projection of the wheel's axis center on the terrain. We show exemplarily the wheel trajectory of the right front leg over a time horizon of one stride duration, which is composed of four splines. The lift-off time t_{lo}, the time at maximum swing height t_{sh}, the touch-down time t_{td}, and the time horizon t_{f} are specified by a fixed gait pattern. The base trajectories are optimized w.r.t. the coordinate frame B whose origin is located at the robot's COM, and whose orientation is equal to that of the frame W.

III. WHEEL TRAJECTORY OPTIMIZATION

We formulate the task of finding the wheel trajectories, i.e., the x, y and z trajectories w.r.t. a wheel coordinate frame W as illustrated in Fig. 3, as a separate quadratic programming (QP) problem for each of the wheels given by

$$\begin{aligned} \underset{\boldsymbol{\xi}}{\text{minimize}} \quad & \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{Q}\boldsymbol{\xi} + \boldsymbol{c}^T\boldsymbol{\xi}, \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{\xi} = \boldsymbol{b},\ \boldsymbol{D}\boldsymbol{\xi} \leq \boldsymbol{f}, \end{aligned} \tag{1}$$

where $\boldsymbol{\xi}$ is the vector of optimization variables. The quadratic objective $\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{Q}\boldsymbol{\xi} + \boldsymbol{c}^T\boldsymbol{\xi}$ is minimized while respecting the linear equality $\boldsymbol{A}\boldsymbol{\xi} = \boldsymbol{b}$ and inequality $\boldsymbol{D}\boldsymbol{\xi} \leq \boldsymbol{f}$ constraints. In the following, the parameterization of the optimization variable is presented, and we introduce each of the objectives, equality constraints and inequality constraints which form the optimization problem.

A. Parameterization of Optimization Variables

We describe the wheel trajectories as a sequence of connected splines. In our implementation, one spline is allocated for each of the two segments where the wheel is in contact with the ground, and two splines are used for describing the trajectory of the wheels in the air. Therefore, the total number of splines for one gait sequence is $n_{\mathrm{s}} = 4$ (see Fig. 3). These two types of trajectory segments, i.e., corresponding to leg in the air and contact, are defined by different parameterizations as described next.

1) Wheel segments in air: We parameterize each coordinate of the wheel trajectory in air as quintic splines. Thus, the position vector at spline segment i is described by

$$\boldsymbol{r}(t) = \begin{bmatrix} \boldsymbol{\eta}^T(t) & \boldsymbol{0}_{1\times6} & \boldsymbol{0}_{1\times6} \\ \boldsymbol{0}_{1\times6} & \boldsymbol{\eta}^T(t) & \boldsymbol{0}_{1\times6} \\ \boldsymbol{0}_{1\times6} & \boldsymbol{0}_{1\times6} & \boldsymbol{\eta}^T(t) \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_{i,x} \\ \boldsymbol{\alpha}_{i,y} \\ \boldsymbol{\alpha}_{i,z} \end{bmatrix} = \boldsymbol{T}(t)\boldsymbol{\xi}_i, \tag{2}$$

where $\boldsymbol{\eta}^T(t) = \begin{bmatrix} t^5 & t^4 & t^3 & t^2 & t & 1 \end{bmatrix}$ and $\boldsymbol{\alpha}_{i,*} \in \mathbb{R}^6$ contains the polynomial coefficients. Here, $t \in [\bar{t}_i, \bar{t}_i + \Delta t_i]$ describes the time interval of spline i with a duration of Δt_i, where $\bar{t}_i$ is the sum of all the previous $(i-1)$ splines' durations (see the example of the fourth spline in Fig. 3). We seek to optimize the polynomial coefficients for all coordinates of spline segment i and hence contain them in the vector $\boldsymbol{\xi}_i = \begin{bmatrix} \boldsymbol{\alpha}_{i,x}^T & \boldsymbol{\alpha}_{i,y}^T & \boldsymbol{\alpha}_{i,z}^T \end{bmatrix}^T \in \mathbb{R}^{18}$.

2) Wheel segments in contact: As shown in our previous work [12], we employ a different parameterization for wheel segments in contact, such that they inherently capture the velocity constraints corresponding to the no-lateral-slip of the wheel. For this purpose, we represent the wheel's velocity in the x coordinate of W, i.e., the rolling direction, as a quadratic polynomial. In contrast, the velocities of the remaining directions are set to zero. Thus, the velocity vector of the i-th spline is

$$\dot{\boldsymbol{r}}(t) = \begin{bmatrix} 1 & t & t^2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_{i,0} \\ \alpha_{i,1} \\ \alpha_{i,2} \end{bmatrix}, \tag{3}$$

and the position vector is obtained by integrating w.r.t. t and adding the initial position $x_i(\bar{t}_i)$ and $y_i(\bar{t}_i)$ of the trajectory as

$$\boldsymbol{r}(t) = \begin{bmatrix} x_i(\bar{t}_i) \\ y_i(\bar{t}_i) \\ 0 \end{bmatrix} + \int_{\bar{t}_i}^{\bar{t}_i+\Delta t_i} \boldsymbol{R}(t\boldsymbol{\omega}_{\mathrm{ref}})\dot{\boldsymbol{r}}(t)\mathrm{d}t = \boldsymbol{T}(\omega_{\mathrm{ref}}, t)\boldsymbol{\xi}_i, \tag{4}$$

where the rotation matrix $\boldsymbol{R}(t\boldsymbol{\omega}_{\mathrm{ref}})$ describes the change in the wheel's orientation caused by the reference yaw rate, i.e., the vector $\boldsymbol{\omega}_{\mathrm{ref}} = \begin{bmatrix} 0 & 0 & \omega_{\mathrm{ref}} \end{bmatrix}^T$. By assuming a constant reference yaw rate ω_{ref} over the optimization horizon, the integration is solved analytically, giving a linear

expression $\boldsymbol{r}(t) = \boldsymbol{T}(\omega_{\mathrm{ref}}, t)\boldsymbol{\xi}_i$ w.r.t. the coefficients $\boldsymbol{\xi}_i = \begin{bmatrix}\alpha_{i,0} & \alpha_{i,1} & \alpha_{i,2} & x_i(\bar{t}_i) & y_i(\bar{t}_i)\end{bmatrix}^T$. Thus, the velocity and acceleration trajectories of spline i are described by $\dot{\boldsymbol{r}}(t) = \dot{\boldsymbol{T}}(\omega_{\mathrm{ref}}, t)\boldsymbol{\xi}_i$ and $\ddot{\boldsymbol{r}}(t) = \ddot{\boldsymbol{T}}(\omega_{\mathrm{ref}}, t)\boldsymbol{\xi}_i$, respectively.

B. Formulation of Trajectory Optimization

To achieve robust locomotion, we deploy an online TO which is executed in a MPC fashion, i.e., the optimization is continuously re-evaluated providing a motion over a time horizon of t_{f} seconds, where t_{f} can be chosen as the stride duration of the locomotion gait.

The complete TO of the wheel trajectories is formulated as a QP problem as follows,

$$
\begin{aligned}
\min_{\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{Q}_{\mathrm{acc}}\boldsymbol{\xi} && \text{acceleration}\\
& + \sum_{k=1}^{N} \left\|\boldsymbol{r}(t_k) - \boldsymbol{r}_{\mathrm{pre}}(t_k + t_{\mathrm{pre}})\right\|^2_{\boldsymbol{W}_{\mathrm{pre}}} \Delta t && \text{previous solution}\\
& \forall t \in [0, t_{\mathrm{f}}] \\
& \text{if leg in contact:} \\
& \quad + \left\|\dot{\boldsymbol{r}}(0) - \boldsymbol{v}_{\mathrm{ref}}\right\|^2_{\boldsymbol{W}_{\mathrm{ref}}} && \text{reference velocity}\\
& \quad + \sum_{k=1}^{N} \left\|r_x(t_k) - r_{x,\mathrm{def}}\right\|^2_{w_{\mathrm{def}}} \Delta t && \text{default position}\\
& \quad \forall t \in [\bar{t}_i, \bar{t}_i + \Delta t_i] \\
& \text{if leg in air:} \\
& \quad + \left\|\boldsymbol{r}_{xy}(t_{\mathrm{td}}) - \boldsymbol{r}_{xy,\mathrm{ref}} - \boldsymbol{r}_{xy,\mathrm{inv}}\right\|^2_{\boldsymbol{W}_{\mathrm{fh}}} && \text{foothold projection}\\
& \quad + \left\|r_z(t_{\mathrm{sh}}) - z_{\mathrm{sh}}\right\|^2_{w_{\mathrm{sh}}}, && \text{swing height}\\
\text{s.t.} \quad & \boldsymbol{r}(0) = \boldsymbol{r}_{\mathrm{init}}, \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_{\mathrm{init}}, \ddot{\boldsymbol{r}}(0) = \ddot{\boldsymbol{r}}_{\mathrm{init}}, && \text{initial state}\\
& \begin{bmatrix}\boldsymbol{r}_i(\bar{t}_i + \Delta t_i)\\ \dot{\boldsymbol{r}}_i(\bar{t}_i + \Delta t_i)\\ \ddot{\boldsymbol{r}}_i(\bar{t}_i + \Delta t_i)\end{bmatrix} = \begin{bmatrix}\boldsymbol{r}_{i+1}(\bar{t}_{i+1})\\ \dot{\boldsymbol{r}}_{i+1}(\bar{t}_{i+1})\\ \ddot{\boldsymbol{r}}_{i+1}(\bar{t}_{i+1})\end{bmatrix}, && \text{spline continuity}\\
& \forall i \in [0, n_{\mathrm{s}} - 1], \\
& \begin{bmatrix}|r_x(t) - r_{x,\mathrm{def}}|\\ |r_y(t) - r_{y,\mathrm{def}}|\\ |r_z(t) - r_{z,\mathrm{def}}|\end{bmatrix} < \begin{bmatrix}x_{\mathrm{kin}}\\ y_{\mathrm{kin}}\\ z_{\mathrm{kin}}\end{bmatrix}, && \text{kinematic limits}\\
& \forall t \in [0, t_{\mathrm{f}}],
\end{aligned}
\tag{5}
$$

where each element is described in more detail in the following sections.

C. Objectives

1) Acceleration minimization: The acceleration $\ddot{\boldsymbol{r}}$ of the entire wheel trajectory is minimized to generate smooth motions and to regularize the optimization problem. The cost term for a wheel in air over the time duration Δt_i of spline i is given by

$$
\frac{1}{2}\boldsymbol{\xi}_i^T \Bigg(\underbrace{2\int_{\bar{t}_i}^{\bar{t}_i + \Delta t_i} \ddot{\boldsymbol{T}}^T(t)\boldsymbol{W}_{i,\mathrm{acc}}\ddot{\boldsymbol{T}}(t)\mathrm{d}t}_{\boldsymbol{Q}_{i,\mathrm{acc}}} \Bigg)\boldsymbol{\xi}_i, \tag{6}
$$

where $\boldsymbol{Q}_{i,\mathrm{acc}} \in \mathbb{R}^{18\times 18}$ is the hessian matrix, and $\boldsymbol{W}_{i,\mathrm{acc}} \in \mathbb{R}^{3\times 3}$ is the corresponding weight matrix. Here, the linear term of (1) is null, i.e., $\boldsymbol{c}_{i,\mathrm{acc}} = \boldsymbol{0}_{18\times 1}$. Similar, for a spline segment i in contact, the hessian matrix, $\boldsymbol{Q}_{i,\mathrm{acc}} \in \mathbb{R}^{5\times 5}$, is obtained by squaring and integrating the acceleration of the wheel trajectory over the time duration Δt_i. The time matrix $\boldsymbol{T}(\omega_{\mathrm{ref}}, t)$, and hence, $\boldsymbol{Q}_{i,\mathrm{acc}}$ is dependent on the reference yaw rate as discussed in (4).

2) Minimize deviations from previous solution: For a TO with high update rates, large deviations between successive solutions can produce quivering motions. To avoid this, we add a cost term that penalizes deviations of kinematic states between consecutive solutions. We penalize the position deviations between the optimization variables from the current solution $\boldsymbol{\xi}$ and the previous solution $\boldsymbol{\xi}_{\mathrm{pre}}$ as

$$
\sum_{k=1}^{N} \left\|\boldsymbol{r}(t_k) - \boldsymbol{r}_{\mathrm{pre}}(t_k + t_{\mathrm{pre}})\right\|^2_{\boldsymbol{W}_{\mathrm{pre}}} \Delta t, \quad \forall t \in [0, t_{\mathrm{f}}], \tag{7}
$$

where $\boldsymbol{r}_{\mathrm{pre}}(t_k + t_{\mathrm{pre}})$ is the position vector of the wheel from the previous solution shifted by the elapsed time t_{pre} since computing the last solution, and $\boldsymbol{W}_{\mathrm{pre}} \in \mathbb{R}^{3\times 3}$ is the corresponding weight matrix. This cost is penalized over the time horizon t_{f} with N sampling points, where t_k is the time at time step k and $\Delta t = t_k - t_{k-1}$. Objectives for minimizing velocity and acceleration deviations are added in a similar formulation.

3) Track reference velocity of wheels in contact: As shown in (3), the velocity along the rolling direction of the wheel trajectory is described by a quadratic polynomial which inherently satisfies the no-slip constraint. To track the reference velocity $\boldsymbol{v}_{\mathrm{ref}}$, we minimize the norm $\left\|\dot{r}_x(0) - v_{x,\mathrm{ref}}\right\|^2_{w_{\mathrm{ref}}}$ which gives

$$
\frac{1}{2}\boldsymbol{\xi}_i^T \underbrace{\left(2w_{\mathrm{ref}}\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\right)}_{\boldsymbol{Q}_{i,\mathrm{ref}}}\boldsymbol{\xi}_i + \underbrace{\left(-2w_{\mathrm{ref}}v_{x,\mathrm{ref}}\boldsymbol{\Gamma}\right)}_{\boldsymbol{c}^T_{i,\mathrm{ref}}}\boldsymbol{\xi}_i, \tag{8}
$$

where $\boldsymbol{\Gamma} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\dot{\boldsymbol{T}}(\omega_{\mathrm{ref}}, 0)$.

4) Minimize deviations from default wheel positions: When a wheel is in contact, differences in heading velocities of the wheels and the base can lead to configurations where the corresponding leg can get extended in the forward or backward direction. To guide the optimizer towards solutions within a desired leg configuration, we minimize the distance of the wheel from a default position $r_{x,\mathrm{def}}$ along the rolling direction x as

$$
\sum_{k=1}^{N} \left\|r_x(t_k) - r_{x,\mathrm{def}}\right\|^2_{w_{\mathrm{def}}} \Delta t, \quad \forall t \in [\bar{t}_i, \bar{t}_i + \Delta t_i], \tag{9}
$$

where w_{def} is the corresponding weight, and the sampling over the i-th contact segment's time duration Δt_i is the same as shown in the paragraph below (7).

5) Foothold projection: The placement of the wheel after a swing phase is crucial for hybrid locomotion (and for legged locomotion in general) because it contributes to maintaining balance and reacting to external disturbances. As shown in (5), the cost term to guide the foothold placement is given by $\left\|\boldsymbol{r}_{xy}(t_{\mathrm{td}}) - \boldsymbol{r}_{xy,\mathrm{ref}} - \boldsymbol{r}_{xy,\mathrm{inv}}\right\|^2_{\boldsymbol{W}_{\mathrm{fh}}}$, where $\boldsymbol{W}_{\mathrm{fh}} \in \mathbb{R}^{2\times 2}$ is

the weight matrix, and $t_{\mathrm{td}} = \bar{t}_i + \Delta t_i$ is the touchdown time of spline segment i in air, i.e., at the end of the spline in air representing the second half of the swing phase (see Fig. 3). The subscript xy indicates that only footholds on the terrain plane are considered, i.e., the z component is given by the height of the terrain estimation.

The position vector $\boldsymbol{r}_{xy,\mathrm{ref}}$ guides the locomotion depending on the reference velocity, which is composed of the linear velocity vector $\boldsymbol{v}_{\mathrm{ref}}$ and the angular velocity vector $\boldsymbol{\omega}_{\mathrm{ref}}$, as

$$\begin{bmatrix} \boldsymbol{r}_{xy,\mathrm{ref}} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_{xy,\mathrm{def}} \\ 0 \end{bmatrix} + (\boldsymbol{v}_{\mathrm{ref}} + \boldsymbol{\omega}_{\mathrm{ref}} \times \boldsymbol{r}_{BW_{xy}})\Delta t_i, \quad (10)$$

where $\boldsymbol{r}_{xy,\mathrm{def}} \in \mathbb{R}^2$ is a specified default wheel position similar to (9), and $\boldsymbol{r}_{BW_{xy}} \in \mathbb{R}^3$ is the position vector from the robot's COM to the projection of the measured wheel position W onto the terrain plane.

Decoupling the locomotion problem into wheel and base TOs requires an additional heuristic to maintain balance. Balancing is achieved by adding a feedback term to the foothold obtained from reference velocities, through an inverted pendulum model [26], [27] given by

$$\boldsymbol{r}_{\mathrm{inv}} = k_{\mathrm{inv}}(\boldsymbol{v}_{BH,\mathrm{ref}} - \boldsymbol{v}_{BH})\sqrt{\frac{h}{g}}, \quad (11)$$

where $\boldsymbol{v}_{BH,\mathrm{ref}} \in \mathbb{R}^3$ and $\boldsymbol{v}_{BH} \in \mathbb{R}^3$ are the reference and the measured velocity between the associated hip and base frame, respectively. Here, h is the height of the hip above the ground, g represents the gravitational acceleration, and k_{inv} is the gain for balancing.

6) Swing height: Similar to the objective in Section III-C3, we guide the wheel TO to match a predefined height. The objective $\|r_z(t_{\mathrm{sh}}) - z_{\mathrm{sh}}\|^2_{w_{\mathrm{sh}}}$ given in (5) can be expanded, with a weight of w_{sh}, to

$$\frac{1}{2}\boldsymbol{\xi}_i^T \underbrace{(2w_{\mathrm{sh}}\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})}_{\boldsymbol{Q}_{i,\mathrm{sh}}}\boldsymbol{\xi}_i + \underbrace{(-2w_{\mathrm{sh}}z_{\mathrm{sh}}\boldsymbol{\Gamma})}_{\boldsymbol{c}^T_{i,\mathrm{sh}}}\boldsymbol{\xi}_i, \quad (12)$$

with $\boldsymbol{\Gamma} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\boldsymbol{T}(t_{\mathrm{sh}})$, and $t_{\mathrm{sh}} = \bar{t}_i + \Delta t_i$ is the time at maximum swing height of spline segment i in air, i.e., at the end of the spline in air representing the first half of the swing phase (see Fig. 3).

Similarly, we set the x and y coordinates of the swing trajectory at maximum swing height to match the midpoint of lift-off and touch-down position.

D. Equality Constraints

1) Initial states: To achieve a reactive behaviour, every optimization is initialized with the current state of the robot. As discussed in (4), the initial position of the wheel segments in contact are set as equality constraints given by

$$\boldsymbol{T}(0)\boldsymbol{\xi}_i = \begin{bmatrix} x_{\mathrm{init}} & y_{\mathrm{init}} & 0 \end{bmatrix}^T, \quad (13)$$

where the initial values x_{init} and y_{init} are the measured positions of the wheel.

If the optimization problem begins with a wheel trajectory in air, we set the initial position, velocity, and acceleration to the measured state of the wheels, i.e., $\boldsymbol{r}(0) = \boldsymbol{r}_{\mathrm{init}}$, $\dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_{\mathrm{init}}$, and $\ddot{\boldsymbol{r}}(0) = \ddot{\boldsymbol{r}}_{\mathrm{init}}$.

2) Spline continuity: We constrain the position, velocity and acceleration at the junction of two consecutive wheel trajectory segments i and $i+1$ in air as

$$\begin{bmatrix} -\boldsymbol{T}_i(\bar{t}_i + \Delta t_i) & \boldsymbol{T}_{i+1}(\bar{t}_{i+1}) \\ -\dot{\boldsymbol{T}}_i(\bar{t}_i + \Delta t_i) & \dot{\boldsymbol{T}}_{i+1}(\bar{t}_{i+1}) \\ -\ddot{\boldsymbol{T}}_i(\bar{t}_i + \Delta t_i) & \ddot{\boldsymbol{T}}_{i+1}(\bar{t}_{i+1}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_i \\ \boldsymbol{\xi}_{i+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{3\times1} \\ \boldsymbol{0}_{3\times1} \\ \boldsymbol{0}_{3\times1} \end{bmatrix}. \quad (14)$$

Junction constraints between air and contact phases are only formulated on position and velocity level. Here, the acceleration is not constrained so that the optimizer accepts abrupt changes in accelerations, allowing lift-off and touch-down events.

E. Inequality Constraints

1) Avoid kinematic limits: To avoid over-extensions of the legs, we keep the wheel trajectories in a kinematic feasible space which is approximated by a rectangular cuboid centered around the default positions defined in (9). As introduced in (5), the kinematic limits x_{kin}, y_{kin}, and z_{kin} are enforced over the full time horizon t_{f} as $|r_x(t_k) - r_{x,\mathrm{def}}| < x_{\mathrm{kin}}$, $|r_y(t_k) - r_{y,\mathrm{def}}| < y_{\mathrm{kin}}$, $|r_z(t_k) - r_{z,\mathrm{def}}| < z_{\mathrm{kin}}$, $\forall k \in [1, .., t_{\mathrm{f}}/\Delta t]$, with a fixed sampling time $\Delta t = t_k - t_{k-1}$ similar to (7).

IV. BASE TRAJECTORY OPTIMIZATION

The online TO of the base motion relies on a ZMP [24]-based optimization, which continuously updates reference trajectories for the free-floating base. Here, we extend the approach shown in our previous work [11], which originates from the motion planning problem of traditional legged robots [17] and does not provide any optimized trajectories for the wheels/feet over a receding horizon. Moreover, the work in [17] only considers the optimization of the footholds. Given the wheel TO in (5), we can generalize the idea of the ZMP to wheeled-legged systems taking into account the trajectories of the wheels over the time horizon t_f.

As shown in Figure 2, the motion planner of the free-floating base is described by a nonlinear optimization problem, which minimizes a nonlinear cost function $\boldsymbol{f}(\boldsymbol{\xi})$ subjected to nonlinear equality $\boldsymbol{c}(\boldsymbol{\xi}) = \boldsymbol{0}$ and inequality constraints $\boldsymbol{h}(\boldsymbol{\xi}) > \boldsymbol{0}$. Here, the vector of optimization variables is composed of the position of the COM $\boldsymbol{r}_{\mathrm{COM}} \in \mathbb{R}^3$ and the yaw-pitch-roll Euler angles of the base $\boldsymbol{\theta} \in \mathbb{R}^3$.

A. Parameterization of Optimization Variables and Formulation of Trajectory Optimization

The trajectories for each DOF of the free-floating base is represented as a sequence of quintic splines, which allows setting position, velocity and acceleration constraints. Thus, the parameterization is formulated similarly to the definition of the wheel trajectories in air given in Section III-A1.

The online TO of the base has a similar structure as the TO described in (5). Cost terms are added to maintain smooth motions and to track the reference velocity. The equality constraints initialize the variables with the current measured state of the base and add junction constraints between consecutive splines. For balancing, we add a ZMP inequality constraint,

which is described in more detail in the next section, since this is the only part of the base optimization problem which is affected by the computed wheel trajectories in Section III. A complete list of each objective and constraint can be obtained in [11].

B. Generalization of ZMP Inequality Constraint

To ensure dynamic stability of the robot, the acceleration of the COM must be chosen so that the ZMP position $\boldsymbol{r}_{\mathrm{ZMP}} \in \mathbb{R}^3$ lies inside the support polygon[3]. This nonlinear inequality constraint is given by

$$\begin{bmatrix} p(t_k) & q(t_k) & 0 \end{bmatrix} \boldsymbol{r}_{\mathrm{ZMP}}(t_k) + r(t_k) \geq 0, \quad \forall t_k \in [0, t_{\mathrm{f}}] \quad (15)$$

where $\boldsymbol{r}_{\mathrm{ZMP}} = \boldsymbol{n} \times \boldsymbol{m}_{\mathrm{gi}} / (\boldsymbol{n}^T \boldsymbol{f}_{\mathrm{gi}})$ [28] and $\boldsymbol{n} \in \mathbb{R}^3$ is the the terrain normal. The *gravito-inertial* wrench [29] is given by $\boldsymbol{f}_{\mathrm{gi}} = m \cdot (\boldsymbol{g} - \ddot{\boldsymbol{r}}_{\mathrm{COM}}) \in \mathbb{R}^3$ and $\boldsymbol{m}_{\mathrm{gi}} = m \cdot \boldsymbol{r}_{\mathrm{COM}} \times (\boldsymbol{g} - \ddot{\boldsymbol{r}}_{\mathrm{COM}}) - \dot{\boldsymbol{l}}_{\mathrm{COM}} \in \mathbb{R}^3$, where m is the mass of the robot, $\boldsymbol{l}_{\mathrm{COM}} \in \mathbb{R}^3$ is the angular momentum of the COM, and $\boldsymbol{g} \in \mathbb{R}^3$ is the gravity vector. In contrast to [11], [17], the line coefficients $\boldsymbol{d}(t) = [p(t)\ q(t)\ r(t)]^T$ that describe an edge of a support polygon depend on the time t, since the contact points of wheeled-legged robots continue to move even when a leg is in contact, unlike conventional legged robots. The ZMP inequality constraint is sampled over the time horizon t_{f} with a fixed sampling time $\Delta t = t_k - t_{k-1}$.

V. EXPERIMENTAL RESULTS AND DISCUSSION

To validate the performance of our hybrid locomotion framework, this section reports on experiments and real-world applications conducted with ANYmal equipped with non-steerable, torque-controlled wheels (see Fig. 1). A video[4] showing the results accompanies this paper.

A. Implementation

The wheel TO, base TO, tracking controller, and state estimator are running on a single PC (Intel i7-7500U, 2.7 GHz, dual-core 64-bit). All computation regarding the autonomy, i.e., perception, mapping, localization, path planning, path following, and object detection, is carried out by three different PCs. The robot is entirely self-contained in terms of computation and perception. As can be obtained in Fig. 2, we run each wheel TO, the base TO, and the WBC in concurrent threads where each optimization reads the last available solutions from its predecessor. Moreover, all optimization problems are run online due to fast solver times.

A hierarchical WBC tracks the computed trajectories in Section III and Section IV by generating torque commands for each actuator and accounting for the full rigid body dynamics including its physical constraints, e.g., the non-holonomic rolling constraint, friction cone, and torque limits [11]. The WBC runs together with state estimation [30] in a 400 Hz loop. Similar to [31], we fuse the inertial measurement unit (IMU) reading and the kinematic measurements from each actuator to acquire the robot's state. Moreover, the frame W in Fig. 3 requires an estimate of the terrain normal. In this work, the robot is locally modeling the terrain as a three-dimensional plane, which is estimated by fitting a plane through the most recent contact locations [11]. The contact state of each leg is determined through an estimation of the contact force, which takes into account the measurements of the motor drives and the full-rigid body dynamics.

We model and compute the kinematics and dynamics of the robot based on the open-source Rigid Body Dynamics Library (RBDL) [32], which uses the algorithms described in [33]. The nonlinear optimization problem in Section IV is solved with a custom sequential quadratic programming (SQP) algorithm, which solves the problem by iterating through a sequence of QP problems. Each QP problem including the optimization problem in Section III is solved using QuadProg++ [34], which internally implements the Goldfarb-Idnani active-set method [35]. To maintain a positive definite Hessian $\boldsymbol{Q}$ in (1) and to ensure the convexity of the resulting QP problem, a regularizer ρ is added to its diagonal elements, e.g., $\rho = 10^{-8}$ as in [17]. The tuning of the cost function in (5) remains a manual task where a single value describes the diagonal elements of the weighting matrices, and one parameter set is provided for all motions shown next.

B. Solver Time of Different Contact Scheduler and Gait Switching

As shown in Table I, the wheel and base optimizations are solved in the order of milliseconds, and a great variety of gaits from driving, i.e., all legs in contact, up to gaits with full-flight phases are possible. Besides, the accompanying video shows manual gait switches between driving and hybrid walking-driving gaits, which can be useful for future works regarding automatic gait switches to reduce the cost of transport (COT) further.

C. Rough Terrain Negotiation

The robot is capable of blind locomotion in a great variety of unstructured terrains, e.g., inclines, steps, gravel, mud, and puddles. Fig. 1 and the accompanying video shows the performance of the robot in these kinds of environments. As depicted in Fig. 4, the robot can overcome blindly steps up to 20 % of its leg length. The obstacle verifies the advantage of our hybrid locomotion framework. In contrast to the related work and our previous work [11], the robot traverses obstacles

TABLE I
TIME HORIZON t_{f} AND OPTIMIZATION TIMES INCLUDING MODEL SETUP FOR DIFFERENT GAITS. THE REPORTED SOLVER TIMES FOR WHEEL TO ARE FOR ONE WHEEL, AND THE HYBRID RUNNING TROT IS A GAIT WITH FULL-FLIGHT PHASES.

Gait	t_{f} / (s)	Wheel TO / (ms)	Base TO / (ms)
Driving	1.7	0.14	6.93
Hybrid walk	2.0	0.81	14.83
Hybrid pace	0.95	0.42	1.88
Hybrid trot	0.85	0.47	2.4
Hybrid running trot	0.64	0.58	5.77

[3]A support polygon is defined by the convex hull of the expected wheels' contact trajectories.

[4]Available at https://youtu.be/ukY0vyM-yfY

without stopping and switching to a pure walking motion. To our best knowledge, this is the first time a robot has demonstrated this level of obstacle negotiation at high speeds, with multiple gaits. Moreover, the locomotion becomes more robust since the framework accounts for possible motions on the ground. The accompanying video shows an instance where the wheel collides with the edge of a step. Our framework is capable of adapting to these scenarios by merely driving over the obstacle.

D. High Speed and Cost of Transport

On flat terrain, the robot achieves a mechanical COT [36] of 0.2 while hybrid trotting at the speed of 2 m/s and the mechanical power consumption is 156 W. The COT is by a factor of two higher than a pure driving gait at the same speed. A comparison to traditional walking and skating with passive wheels [36] shows that the COT is lower by 42 % w.r.t. the traditional trotting gait and by 9 % w.r.t. skating motions.

E. DARPA Subterranean Challenge: Tunnel Circuit

The first DARPA Subterranean Challenge, the Tunnel Circuit, was held close to Pittsburgh in the NIOSH mine. The main objective was to search, detect, and provide autonomously spatially referenced locations of artifacts inside the underground mine. The wheeled version of ANYmal participated in two runs as part of the CERBERUS team [37] alongside flying and other mobile platforms. Moreover, the wheeled quadrupedal robot was deployed next to the traditional version of ANYmal without wheels.

As depicted in the lower images of Fig. 1, the terrain consisted of hilly, bumpy, and muddy terrain and in some parts of the mine, the robot needed to cross puddles. Throughout both runs, the robot locomoted the terrain with a hybrid trot. In the first run, the wheeled version of ANYmal traversed 70 m without significant issues, and the robot successfully reported the correct location of one artifact. In the end, however, one of the wheels started slipping on the muddy terrain before the fall. As can be seen in the accompanying video, the robot managed to balance after the first slip because of the foothold adaptation of the inverted pendulum model in (11). The mechanical design was improved after the first run by adding a chain around the wheels to increase the friction coefficient while traversing the mud (see the right middle image of Fig. 1). Fig. 5 shows the desired trajectories of the COM and wheels for a few meters of the second run. Here, it can be seen that the robot executes a hybrid trotting gait since, during ground contact, the wheel moves along its rolling direction. Despite the challenging environment, the hybrid locomotion framework enabled the robot to travel for more than 100 m.

Due to the time limitation of the challenge, the speed of mobile platforms becomes an essential factor. Most of the wheeled platforms shown from the other competing teams were faster than our traditional legged robot by a factor of two or more. The upcoming Urban Circuit of the Subterranean Challenge includes stairs and other challenging obstacles. Therefore, we believe, only a wheeled-legged robot is capable of combining speed and versatility. At the Tunnel Circuit, the

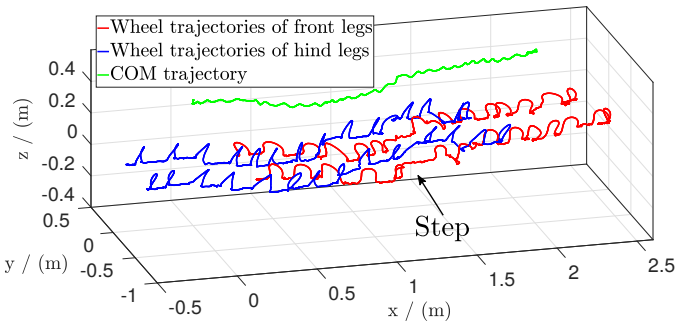

Fig. 4. Measured COM and wheel trajectories of ANYmal over a step while hybrid trotting, as depicted in the upper images of Fig. 1. The three-dimensional plot shows the wheel trajectories of the front legs (red line), the wheel trajectories of the hind legs (blue line), and the COM trajectory (green line) w.r.t. the inertial frame, which is initialized at the beginning of the run.

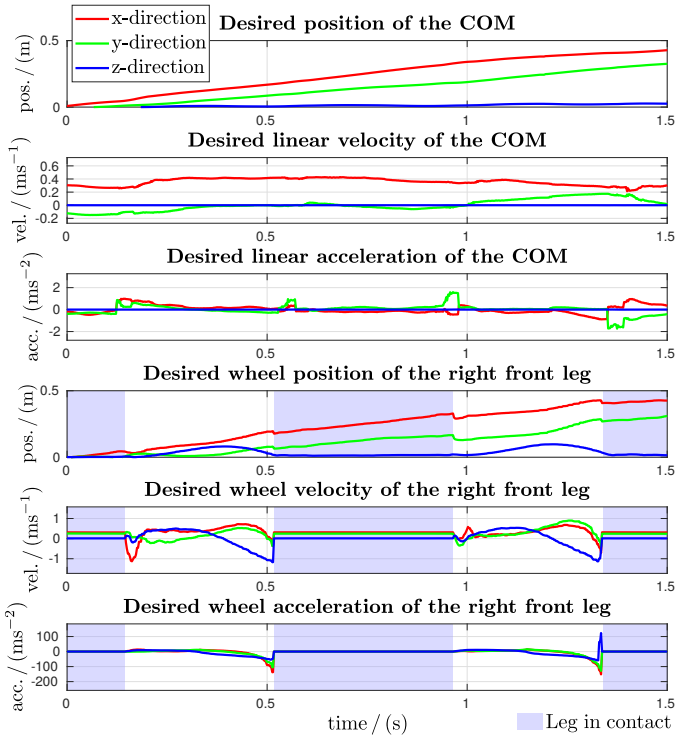

Fig. 5. Desired COM and wheel trajectories of ANYmal at the DARPA Subterranean Challenge. The robot, ANYmal, is autonomously locomoting with a hybrid driving-trotting gait during the second scoring run. The environment is a wet, inclined, muddy, and rough underground mine, as depicted in the lower images of Fig. 1. Despite the challenging terrain, the robot manages to explore fully autonomously the mine for more than 100 m. The plots show the desired motions for approximately two stride durations. Due to the fast update rates of the TO problems and reinitialization of the optimization problem with the measured state, the executed trajectories are almost identical to the desired motion shown here.

wheeled version of ANYmal traversed with an average speed of 0.5 m/s, which was more than double the average speed of the traditional legged system. Our chosen speed was limited by the update frequency of our mapping approach or otherwise could have traversed the entire terrain with much higher speeds without any loss in agility. On the whole, the performance validation for real-world applications is satisfying, and a direct comparison with the traditional ANYmal reveals the advantages of wheeled-legged robots.

VI. CONCLUSIONS

This work presents an online TO generating hybrid walking-driving motions on a wheeled quadrupedal robot. The optimization problem is broken down into wheel and base trajectory generation. The two independent TOs are synchronized to generate feasible motions by time sampling the prior generated wheel trajectories, which form the support polygons of the ZMP inequality constraint of the base TO. The presented algorithm makes the locomotion planning for high dimensional wheeled-legged robots more tractable, enables us to solve the problem in real-time on-board in a MPC fashion, and increases the robustness in the robot's locomotion against unforeseen disturbances.

To the best of our knowledge, this is the first time that a hybrid walking-driving robot is deployed for real-world missions at one of the biggest robotics competition. In future work, we plan to incorporate the optimization of the gait timings to enable automatic switching between pure driving and hybrid walking-driving. As shown in our work, an automated way of choosing when to lift a leg can increase the speed and robustness of the locomotion.

REFERENCES

[1] C. D. Bellicoso, M. Bjelonic, L. Wellhausen, K. Holtmann, F. Günther, M. Tranzatto, P. Fankhauser, and M. Hutter, "Advances in real-world applications for legged robots," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1311–1326, 2018.

[2] P. Eckert, A. Spröwitz, H. Witte, and A. J. Ijspeert, "Comparing the effect of different spine and leg designs for a small bounding quadruped robot," in *IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 3128–3133.

[3] J. A. Nyakatura, K. Melo, T. Horvat, K. Karakasiliotis, V. R. Allen, A. Andikfar, E. Andrada, P. Arnold, J. Lauströer, J. R. Hutchinson, *et al.*, "Reverse-engineering the locomotion of a stem amniote," *Nature*, vol. 565, no. 7739, p. 351, 2019.

[4] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Blösch, *et al.*, "Anymal-toward legged robots for harsh environments," *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, 2017.

[5] W. Reid, F. J. Pérez-Grau, A. H. Göktoğan, and S. Sukkarieh, "Actively articulated suspension for a wheel-on-leg rover operating on a martian analog surface," in *IEEE Int. Conf. on Robotics and Automation*, 2016, pp. 5596–5602.

[6] P. R. Giordano, M. Fuchs, A. Albu-Schaffer, and G. Hirzinger, "On the kinematic modeling and control of a mobile platform equipped with steering wheels and movable legs," in *IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 4080–4087.

[7] F. Cordes, C. Oekermann, A. Babu, D. Kuehn, T. Stark, F. Kirchner, and D. R. I. C. Bremen, "An active suspension system for a planetary rover," in *Proceedings of the Int. Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, 2014, pp. 17–19.

[8] M. Giftthaler, F. Farshidian, T. Sandy, L. Stadelmann, and J. Buchli, "Efficient kinematic planning for mobile manipulators with non-holonomic constraints using optimal control," in *IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 3411–3417.

[9] A. Suzumura and Y. Fujimoto, "Real-time motion generation and control systems for high wheel-legged robot mobility," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3648–3659, 2014.

[10] C. Grand, F. Benamar, and F. Plumet, "Motion kinematics analysis of wheeled–legged rover over 3d surface with posture adaptation," *Mechanism and Machine Theory*, vol. 45, no. 3, pp. 477–495, 2010.

[11] M. Bjelonic, C. D. Bellicoso, Y. de Viragh, D. Sako, F. D. Tresoldi, F. Jenelten, and M. Hutter, "Keep rollin' - whole-body motion control and planning for wheeled quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2116–2123, 2019.

[12] Y. de Viragh, M. Bjelonic, C. D. Bellicoso, F. Jenelten, and M. Hutter, "Trajectory optimization for wheeled-legged quadrupedal robots using linearized zmp constraints," in *IEEE Robotics and Automation Letters*, 2019.

[13] A. Laurenzi, E. M. Hoffman, and N. G. Tsagarakis, "Quadrupedal walking motion and footstep placement through linear model predictive control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 2267–2273.

[14] T. Klamt and S. Behnke, "Planning hybrid driving-stepping locomotion on multiple levels of abstraction," in *IEEE Int. Conf. on Robotics and Automation*, 2018, pp. 1695–1702.

[15] G. Bellegarda and K. Byl, "Trajectory optimization for a wheel-legged system for dynamic maneuvers that allow for wheel slip," in *under review for IEEE Conf. on Decision and Control (CDC)*, 2019.

[16] M. Geilinger, R. Poranne, R. Desai, B. Thomaszewski, and S. Coros, "Skaterbots: Optimization-based design and motion synthesis for robotic creatures with legs and wheels," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, p. 160, 2018.

[17] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2261–2268, 2018.

[18] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback mpc for torque-controlled legged robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019.

[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] J. Carius, R. Ranftl, V. Koltun, and M. Hutter, "Trajectory optimization for legged robots with slipping motions," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 3013–3020, 2019.

[21] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 1–9.

[22] Q. Nguyen, M. J. Powell, B. Katz, J. Di Carlo, and S. Kim, "Optimized jumping on the mit cheetah 3 robot," in *IEEE Int. Conf. on Robotics and Automation*, 2019, pp. 7448–7454.

[23] O. A. V. Magaña, V. Barasuol, M. Camurri, L. Franceschi, M. Focchi, M. Pontil, D. G. Caldwell, and C. Semini, "Fast and continuous foothold adaptation for dynamic locomotion through cnns," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2140–2147, 2019.

[24] M. Vukobratović and B. Borovac, "Zero-moment point – thirty five years of its life," *Int. journal of humanoid robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[25] F. Farshidian, "Planning and control in face of uncertainty with applications to legged robots," Ph.D. dissertation, ETH Zurich, 2017.

[26] C. Gehring, S. Coros, M. Hutter, C. D. Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. Hoepflinger, *et al.*, "Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot," *IEEE Robotics & Automation Magazine*, vol. 23, no. 1, pp. 34–43, 2016.

[27] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.

[28] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. center of pressure-zero moment point," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 34, no. 5, pp. 630–637, 2004.

[29] S. Caron, Q.-C. Pham, and Y. Nakamura, "Zmp support areas for multicontact mobility under frictional constraints," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 67–80, 2017.

[30] M. Bloesch, M. Burri, H. Sommer, R. Siegwart, and M. Hutter, "The two-state implicit filter recursive estimation for mobile robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 573–580, 2018.

[31] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and imu," *Robotics*, vol. 17, pp. 17–24, 2013.

[32] M. Felis. Rigid Body Dynamics Library. [Online]. Available: https://bitbucket.org/rbdl/rbdl/src/default/

[33] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[34] L. D. Gasper. QuadProg++. [Online]. Available: http://quadprog.sourceforge.net/

[35] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical programming*, vol. 27, no. 1, pp. 1–33, 1983.

[36] M. Bjelonic, C. D. Bellicoso, M. E. Tiryaki, and M. Hutter, "Skating with a force controlled quadrupedal robot," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 7555–7561.

[37] CERBERUS. [Online]. Available: https://www.subt-cerberus.org/