Trajectory Optimization for Hybrid Wheeled-Legged Robots in Challenging Terrain*

Vivian Suzano Medeiros and Marco Antonio Meggiolaro

Pontifical Catholic University of Rio de Janeiro, Gávea, Rio de Janeiro, RJ, 22451-900, Brazil
viviansuzano@puc-rio.br
meggi@puc-rio.br

Abstract. Wheeled-legged robots are a promising solution for agile locomotion in challenging terrain, combining the speed of the wheels with the ability of the legs to cope with unstructured environments. This paper presents a trajectory optimization framework that allows wheeled-legged robots to navigate in challenging terrain, e.g., steps, slopes, gaps, while negotiating these obstacles with dynamic motions. The framework generates the robot's base motion as well as the wheels' positions and contact forces along the trajectory, accounting for the terrain map and the dynamics of the robot. The knowledge of the terrain map allows the optimizer to generate feasible motions for obstacle negotiation in a dynamic manner, at higher speeds. To take full advantage of the hybrid nature of wheeled-legged robots, driving and stepping motions are both considered in a single planning problem that can generate trajectories with purely driving motions or hybrid driving-stepping motions. The optimization is formulated as a Nonlinear Programming Problem (NLP) employing a phase-based parametrization to optimize over the wheels' motion and contact forces. The reference trajectories are tracked by a hierarchical whole-body controller that computes the torque actuation commands for the robot. The motion plans are verified on the quadrupedal robot ANYmal equipped with non-steerable torque-controlled wheels in simulations and experimental tests. Agile hybrid motions are demonstrated in simulations with discontinuous obstacles, such as floating steps and gaps, at an average speed of 0.75 m/s.

Keywords: Wheeled-legged robots · Trajectory optimization · Challenging Terrain.

Student Level: PhD
Thesis defended in July 29th, 2020.
Submission to CTDR.

* This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001.

1 Introduction

The number of applications for autonomous ground robots that require navigation in rough terrain has substantially increased in the last years, including planetary exploration, farming, industrial inspection, and search and rescue. For such scenarios, wheeled-legged robots offer a versatile solution, combining the advantages of both legged and wheeled locomotion, which allows the robot to cope with challenging environments at higher speeds. Tasks where rapid execution time is essential would greatly benefit from such systems. For this reason, this work focuses on developing a Trajectory Optimization (TO) framework for wheeled-legged robots that enables dynamic locomotion in challenging terrain.

In the field of planning for wheeled-legged robots, most of the previous work focuses on reactive locomotion in a quasi-static condition. The most common examples are extra-planetary rovers [8, 4, 15], which employ a purely reactive controller that can adapt to terrain variations by maintaining a desired base pose. These controllers are typically able to execute statically-stable driving motions at low speeds, where the legs act as a sophisticated active suspension system and are not used for stepping motions.

The motion framework presented in [11, 12] for the *CENTAURO* robot switches between stepping or driving based on the terrain complexity, employing a kinematic approach for locomotion. Experimental results show the robot overcoming obstacles like stones, steps and gaps with slow static maneuvers. Such approaches do not consider solutions where the robot uses its wheels and legs simultaneously, which limits their ability to overcome obstacles compared to considering the whole-body in a single planning problem. On the other hand, the motion planner presented by [10] solves the whole-body planning problem combining driving and stepping motions, but focuses on generating kinematically feasible motions for heavy wheeled-legged vehicles performing slow maneuvers.

Dynamic motion generation has been shown for the robot ANYmal equipped with actuated wheels in [5, 2, 3], where the reference trajectories for the robot's base are computed by a Zero-Moment Point (ZMP) optimization and tracked by a hierarchical whole-body controller (WBC). Experimental tests show the robot performing separated driving and walking motions [2], and hybrid driving-walking motions [3] over rough terrain. However, the motion planner uses a flat terrain assumption, which violates the validity of the ZMP model when moving over non-flat terrain and renders the approach not amenable for terrains with steep steps or discontinuous obstacles.

The framework *Skaterbots* [7] shows a general TO framework for wheeled-legged robots that optimizes over several types of hybrid motions by solving an NLP. However, it assumes the robot is moving on flat ground and it does not include any terrain information. The TO framework presented in [1] for the wheeled-legged robot *Robosimian* generates dynamic hybrid driving-walking motions, but for passive wheels and limited to flat terrain.

All of these approaches, even though can generate dynamic motions, do not take into account terrain information and the flat terrain assumption makes them not very well suited for cases where the terrain contains abrupt obstacles.

In contrast, a recent work presented in [17] introduces a hierarchical control framework for the *Pholus* robot designed to perform hybrid locomotion in uneven terrain, taking into account the terrain height changes in the motion planning. Experimental results shows the robot overcoming thin obstacles and steps with statically stable hybrid motions at low speeds.

In general, previous work in motion planning and control for wheeled-legged robots have focused either on statically stable navigation in challenging terrain at low speeds (less than 0.15 m/s), or on dynamic locomotion employing a flat terrain assumption. To bridge this gap, the main contribution of this work is introducing a motion planning framework for wheeled-legged robots capable of negotiating challenging terrain with dynamic motions. The proposed TO framework optimizes over the robot's base motion (position and orientation), the wheels' contact positions and contact forces in a single planning problem, accounting for the terrain map and the robot's dynamics. This combination allows the robot to traverse a variety of challenging terrain with dynamic motions that could not be generated without taking into account the terrain information. In addition, the framework allows for purely driving motions and hybrid driving-stepping motions, that can overcome discontinuities in the terrain profile by performing simultaneous driving and stepping. The approach is validated in both simulations and experiments with the ANYmal robot equipped with actuated torque-controlled wheels. We show the robot performing hybrid dynamic motions to overcome floating steps and gaps at an average speed of 0.75 m/s. To the best of our knowledge, such a fast negotiation of discontinuous obstacles using hybrid motions has not been shown before.

2 Trajectory Optimization

Planning hybrid motions for wheeled-legged robots is a challenging task that involves several constraints to be fulfilled by the planned trajectories. Similarly to legged systems with point feet, the wheels can either be in contact with the ground (contact phase) or not (flight phase). The main difference is that the velocity of the end-effector is no longer forced to zero when in contact with the ground and the rolling direction of the wheels is constrained to ensure consistency with wheeled locomotion. In addition, the forces that move the robot can only be created when the wheel is in contact with the terrain. As a consequence, the wheel that is being lifted off does not produce any forces.

The complete TO formulation for wheeled-legged robots is summarized in Fig. 1. The decision variables are the base's motion (position and orientation), the wheels' contact positions and the contact forces. The high-level user inputs are the robot's initial and final state, the total time duration T of the trajectory and the contact schedule for the motion, which indicates the sequences and durations of the contact phases for the wheels. Fig. 2 depicts the coordinate frames of the robot used in the formulation. For all the variables, the right superscript denotes a component of the vector and the left superscript indicates the coordinate frame.

$$
\begin{aligned}
\text{find} \quad & {}^{I}\boldsymbol{r}(t) \in \mathbb{R}^3 && \text{(CoM position)} \\
& {}^{I}\boldsymbol{\theta}(t) \in \mathbb{R}^3 && \text{(CoM Euler angles)} \\
& \text{for every wheel } i: \\
& \quad {}^{I}\boldsymbol{p}_i(t) \in \mathbb{R}^3 && \text{(wheels' motion)} \\
& \quad {}^{I}\boldsymbol{f}_i(t) \in \mathbb{R}^3 && \text{(wheels' forces)} \\
\text{s.t.} \quad & [{}^{I}\boldsymbol{r}, {}^{I}\boldsymbol{\theta}](0) = [{}^{I}\boldsymbol{r}_0, {}^{I}\boldsymbol{\theta}_0] && \text{(initial state)} \\
& [{}^{I}\boldsymbol{r}, {}^{I}\boldsymbol{\theta}](T) = [{}^{I}\boldsymbol{r}_g, {}^{I}\boldsymbol{\theta}_g] && \text{(goal state)} \\
& \boldsymbol{F}_d({}^{I}\boldsymbol{r}, {}^{I}\boldsymbol{\theta}, {}^{I}\boldsymbol{p}_i, {}^{I}\boldsymbol{f}_i) = \boldsymbol{0} && \text{(dynamic model)} \\
& \text{for every wheel } i: \\
& \quad {}^{I}\boldsymbol{p}_i(t) \in \mathcal{R}_i({}^{I}\boldsymbol{r}(t), {}^{I}\boldsymbol{\theta}(t)) && \text{(kinematic constraint)} \\
& \quad \text{if wheel } i \text{ is in contact:} \\
& \quad\quad {}^{I}p_i^z(t) = h_{terrain}({}^{I}\boldsymbol{p}_i^{x,y}(t)) && \text{(terrain height)} \\
& \quad\quad {}^{C_i}\boldsymbol{f}_i^z(t) > 0 && \text{(normal force)} \\
& \quad\quad || {}^{C_i}\boldsymbol{f}_i^x(t)|| \leq f_{max} && \text{(maximum torque)} \\
& \quad\quad {}^{I}\boldsymbol{f}_i(t) \in \mathcal{F}(\mu, \boldsymbol{n}, {}^{I}\boldsymbol{p}_i^{x,y}(t)) && \text{(friction cone)} \\
& \quad\quad {}^{C_i}\dot{p}_i^y(t) = 0 && \text{(rolling constraint)} \\
& \quad \text{if wheel } i \text{ is not in contact:} \\
& \quad\quad {}^{I}\boldsymbol{f}_i(t) = \boldsymbol{0} && \text{(no force)} \\
& \quad\quad {}^{I}p_i^z(t) > h_{terrain}({}^{I}\boldsymbol{p}_i^{x,y}(t)) && \text{(no contact)}
\end{aligned}
$$

Fig. 1. Decision variables and constraints of the TO formulation. The set of constraints on the wheels' variables depends on the contact state of the wheels.

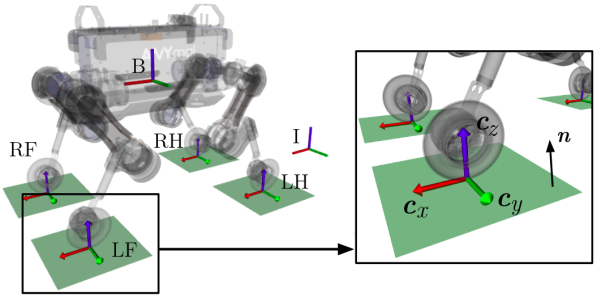

Fig. 2. Coordinate frames for the motion planning: I denotes the inertial frame and B denotes the base frame, attached to the robot's CoM. On the right, a detailed view of the wheel's contact frame C_i. The axis $\boldsymbol{c}_z$ is aligned with the terrain normal $\boldsymbol{n}$ and the $\boldsymbol{c}_x$ axis is aligned with the rolling direction of the wheel. For the wheels, the first letter indicates left (L) or right (R) and the second indicates front (F) or hind (H).

2.1 Optimization Variables Parametrization

In our approach, we employ a *Direct Collocation* method [9], where the continuous problem is transcribed into an NLP problem by optimizing over the decision variables in discrete times sampled along the trajectory, called *nodes*. The continuous motion is then obtained by sequences of third-order polynomials using the Hermite parametrization, which ensures continuous derivatives at the polynomial junctions. For the base motion, the nodes are sampled at a fixed time interval ΔT and the constraints are enforced on all nodes. The difficulty resides in the formulation for the wheels variables (force and motion), since the set of constraints changes depending on the contact state of the wheels.

To overcome this problem, a phase-based parameterization based on [19] is employed, where each wheel motion is treated individually, alternating between contact and swing phases. The sequence and durations of each phase are specified in advance by the user and phase-specific constraints can be enforced directly at the polynomial junctions. Fig. 3 illustrates the wheels parametrization for performing a hybrid driving-walking motion on flat terrain. Each phase is represented by a sequence of Hermite polynomials and the number of polynomials changes depending on the contact state. Each phase has a fixed duration ΔT_j, for $j = 1 \,..\, n_p$, where n_p is the total number of phases.

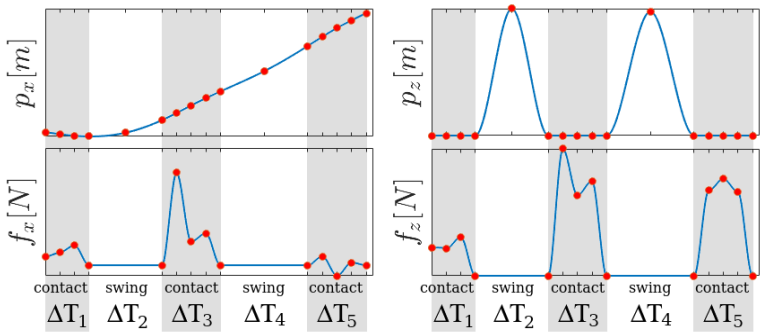

Fig. 3. Phase-based parametrization for the wheels' contact positions and forces. Nodes (red dots) in the gray area are contact nodes and nodes in the white area are swing nodes. Graphs p_x and p_z are the x and z dimension of the wheel contact point trajectory, while f_x and f_z represent the x and z dimension of the wheel contact force.

The contact phases have nodes sampled at a fixed interval Δt_d. It is important that this time interval is not very large, so it is possible to generate feasible trajectories even for gaits with long contact phases, such as driving motions. Both the wheels' positions and forces are allowed to have non-zero values and are parametrized with the same number of nodes.

In contrast, swing phases are mainly characterized by no contact between the wheel and the ground, which implies zero contact forces. This is ensured in the parametrization by fixing a constant zero value for the forces in all swing phases, as implemented in [19]. For the wheels' motion, at least two polynomials

are necessary, so the wheel can be lifted off and then lowered back down, as depicted in the white areas of the p_z graph in Fig. 3.

Several different gait patterns can be used for defining the sequences and durations of the contact phases. Fig. 4 shows some of the gaits used for generating the trajectories in this work, where the contact phases are indicated with colors and swing phases with white spaces. Tests were performed with both statically and dynamically stable gaits, including gaits with full flight phases.

Fig. 4. Some examples of gaits used for hybrid trajectory generation: (a) driving, (b) hybrid gallop, (c) hybrid running trot, (d) hybrid bounding.

2.2 Dynamic and Kinematic Constraints

The robot's dynamic model used for the TO is a Single Rigid Body Dynamics (SRBD) model, in which the robot is approximated by a single rigid-body with mass and inertia located at the robots CoM, as depicted in Fig. 5(a). This model assumes that the mass and inertia of the legs are negligible compared to the robot's base which is reasonable for most legged robots, since each leg is up to an order of magnitude lighter than the base. With this assumption, the robot's CoM linear acceleration $\ddot{\boldsymbol{r}}(t) \in \mathbb{R}^3$ and angular acceleration $\dot{\boldsymbol{\omega}}(t) \in \mathbb{R}^3$ are given by

$$\begin{aligned} m\,{}^I\ddot{\boldsymbol{r}}(t) &= \sum_{i=1}^{4} {}^I\boldsymbol{f}_i(t) - m\,{}^I\boldsymbol{g}, \\ \boldsymbol{I}\,{}^I\dot{\boldsymbol{\omega}}(t) + {}^I\boldsymbol{\omega}(t) \times \boldsymbol{I}\,{}^I\boldsymbol{\omega}(t) &= \sum_{i=1}^{4} {}^I\boldsymbol{f}_i(t) \times ({}^I\boldsymbol{r}(t) - {}^I\boldsymbol{p}_i(t)), \end{aligned} \tag{1}$$

where ${}^I\boldsymbol{\omega}(t) \in \mathbb{R}^3$ represents the angular velocity of the robot's base in the world frame, m denotes the robot's mass, $\boldsymbol{g} \in \mathbb{R}^3$ the gravity vector and $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ the inertia matrix of the robot, computed around the nominal stance position.

As for the kinematic constraints, the wheels' positions are constrained to remain within a feasible workspace that moves together with the robot's base, approximated by a parallelepiped with fixed size located on the nominal position of the wheel relative to the robot's base, as depicted in Fig. 5(b). The constraint is given by

$$-\boldsymbol{b} \;\leq\; \boldsymbol{R}_{BI}(\boldsymbol{\theta}(t))({}^I\boldsymbol{p}_i(t) - {}^I\boldsymbol{r}(t)) - {}^B\boldsymbol{p}_{in} \;\leq\; \boldsymbol{b}, \tag{2}$$

where $\boldsymbol{R}_{BI} \in \mathbb{R}^{3\times3}$ is the rotation matrix from the inertial frame to the base frame, ${}^B\boldsymbol{p}_{in} \in \mathbb{R}^3$ is the nominal position of the i^{th} wheel in the base frame and $\boldsymbol{b} = \begin{bmatrix} b_x & b_y & b_z \end{bmatrix}^T$ is the vector of the parallelepiped dimensions.

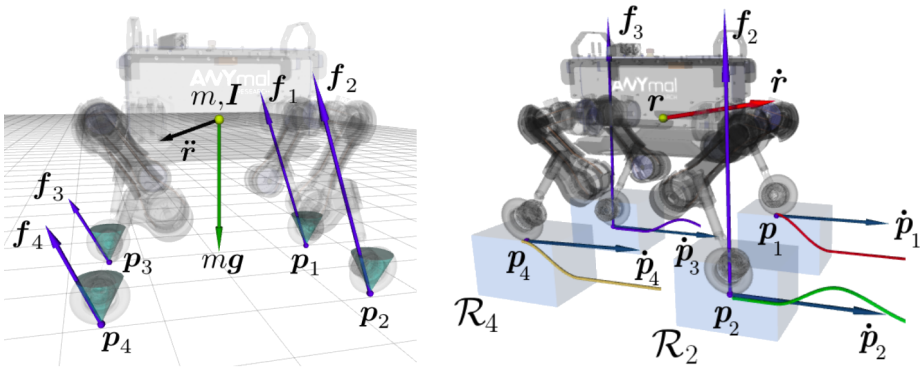

Fig. 5. (a) Single rigid body representation of a wheeled quadrupedal robot. (b) The NLP decision variables for a wheeled quadrupedal robot performing a hybrid running trot. The wheels' trajectories for are indicated each with a different color, matching the gait pattern graphs. The parallelepiped $\mathcal{R}_i$ indicates the feasible workspace for the i^{th} wheel.

2.3 Force Constraints

During contact phases, the following constraints are enforced on the wheels' force profiles: unilateral constraint, friction cone and maximum wheel torque. They are all defined in the following equations, for which $\mathcal{S}_i$ indicates the set of time intervals in which the i^{th} wheel is in contact (stance) phase.

The unilateral constraint ensures the contact force always pushes into the terrain, which is equivalent to constrain the component of the contact force orthogonal to the terrain to be positive. This translates as follows:

$$^{C_i}f_i^z(t \in \mathcal{S}_i) > 0 \tag{3}$$

where $^{C_i}f_i^z(t)$ is the z component of the contact force on the i^{th} wheel expressed in the i^{th} contact frame.

For ensuring no slippage while driving, the contact forces are constrained to remain inside the Coulomb friction cone defined by the terrain friction coefficient μ, depicted in Fig. 5(a). In our implementation, the friction cone is approximated by a friction pyramid, which makes the constraint linear and thus, speeds up the computation. The constraint is given by

$$-\mu\, ^{C_i}f_i^z(t \in \mathcal{S}_i) \leq {}^{C_i}f_i^x(t \in \mathcal{S}_i) \leq \mu\, ^{C_i}f_i^z(t \in \mathcal{S}_i) \tag{4}$$

$$-\mu\, ^{C_i}f_i^z(t \in \mathcal{S}_i) \leq {}^{C_i}f_i^y(t \in \mathcal{S}_i) \leq \mu\, ^{C_i}f_i^z(t \in \mathcal{S}_i) \tag{5}$$

Lastly, the torque limits from the wheel's actuators are imposed by limiting the traction force $^{C_i}f_i^x$ to a maximum value, as such:

$$-\tau_{max}/r_w \leq {}^{C_i}f_i^x(t \in \mathcal{S}_i) \leq \tau_{max}/r_w, \tag{6}$$

where τ_{max} is the maximum allowed torque by the wheels' motors and r_w is the wheel's radius.

2.4 Motion Constraints

When in contact, the wheel can have a non-zero speed or acceleration, but limited to the rolling direction of the wheels for consistency with driving locomotion. This is enforced by constraining the y-velocity of the wheels' contact point to be zero:

$$^{C_i}\dot{p}_i^y(t \in \mathcal{S}_i) = 0. \tag{7}$$

where $^{C_i}p_i^y(t)$ is the y component of the i^{th} wheel contact position expressed in the i^{th} contact frame. It is important that this constraint is only enforced during contact phases, since the wheel is allowed to move in all directions when in swing phase.

In addition, contact between the wheel and the ground must be ensured during contact phases, which is enforced by

$$^{I}p_i^z(t \in \mathcal{S}_i) = h_{terrain}(^{I}\boldsymbol{p}_i^{x,y}(t)) \tag{8}$$

where $h_{terrain}$ is the continuous 2.5D height map of the terrain [6].

3 Results

This section presents the implementation and testing of several motions generated by the TO framework. The hybrid motion plans are validated in physical simulations[1] with the ANYmal robot equipped with non-steerable wheels in different terrains, including a 0.25 m gap and a floating step with a 0.2 m height. These scenarios are examples of situations where stepping is required and for which hybrid motions are particularly well suited. Dynamic driving motions are demonstrated in real-world experiments[2] with ANYmal negotiating different steep steps in our previous paper [14].

3.1 Implementation

The TO framework is implemented in C++ using the Ifopt [20] interface for the interior-point method solver Ipopt [18]. All derivatives are provided to the solver analytically, which significantly improves its performance. Given the contact schedule, the height map and the goal state, the TO framework computes the reference trajectories for the base and the wheels, as well as the wheels' contact forces. The planned trajectories are tracked by the WBC developed in [2], which computes the actuation torques for the joints and the wheels at a 400 Hz frequency while accounting for several constraints, such as actuator limitations, friction cone and the nonholonomic rolling constraints. Fig. 6 gives an overview of the motion planning framework.

For the trajectory optimization, the kinematic and dynamic constraints are enforced every 0.1 s, which is enough to ensure the feasibility of the motion

[1] A video showing the dynamic hybrid motions is available in `https://drive.google.com/file/d/1J5-Hzk6R8gQzoxWYzpbZPkTHZhOhYCSv/view`

[2] A video showing the experiments is available in `https://youtu.be/DlJGFhGS3HM`

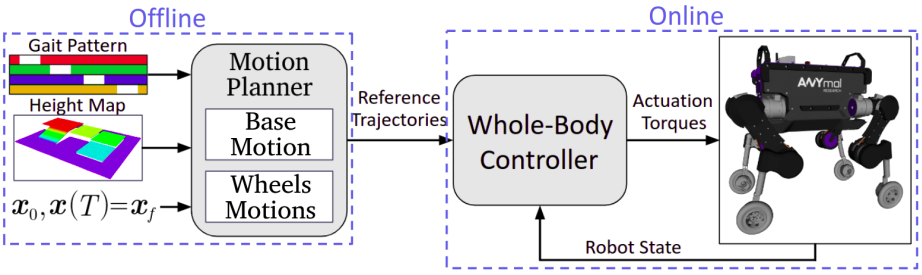

Fig. 6. Overview of the motion planning framework for wheeled-legged robots.

plans. On contact phases, the force and motion constraints are enforced every 0.1 s. For swing phases, two polynomials were used for describing the motion in most cases, but three polynomials were also used to improve the convergence for more difficult cases, such as going up a floating step. The TO problem consists of a non-linear non-convex optimization, which is a challenging task to solve. The solver computation time depends on the complexity of the terrain and the optimization parameters, but remained in average 2.1 times[3] shorter than the planning horizon.

For the optimization, the base and the wheel's trajectories are initialized with a linear interpolation between the initial and final position of the robot, assuming the average speed during the entire motion. Similar to the formulation presented in [14], we avoid using a cost function in the optimization to keep a low computational cost, so the objective is simply to find a feasible solution. Lastly, the continuous 2.5D height map of the terrain required for the motion planning was analytically defined for each of the terrains.

3.2 Simulations

The simulations are carried out in the robot simulation environment Gazebo [13] with ODE [16] as the physics engine. To ensure realistic results, we use the full rigid body dynamics of the robot, accounting for the torque limits of the actuators.

The robot's gained ability to traverse discontinuous obstacles with dynamic hybrid motions is demonstrated in Fig. 7, in which the robot is required to cross a gap with a 0.25 m width at an average speed of 0.75 m/s. The gait used for this maneuver is a hybrid bounding gait, shown in Figure 4(d). The robot starts lifting both front legs prior to the obstacle while the hind legs continue to drive forward. When hopping with the front wheels, the robot moves the base backwards to increase the support forces on the hind wheels and maintain stability. This behavior is only possible because we take into account the dynamics of the robot during the motion planning. Once the front wheels are on the ground, the hind wheels are moved to the other side of the gap, completing the maneuver.

[3] The times stated in this work for the TO computation times were obtained on a 2.7 GHz dual-core Intel Core i7 laptop.

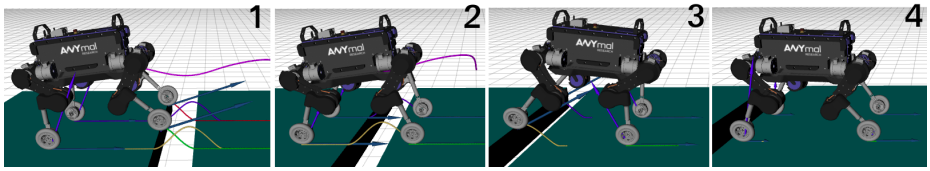

Fig. 7. The ANYmal robot performing a dynamic hybrid bounding gait to cross a 0.25 m gap. The light blue arrows on the wheels are the contact forces and the dark blue arrows are the wheels' linear velocity.

Fig. 8 depicts the desired motions compared with the measured positions obtained from the simulations, confirming the successful tracking of the WBC. The WBC was able to track the desired motions with an average Root-Mean-Square-Error (RMSE) of 20.1 mm.

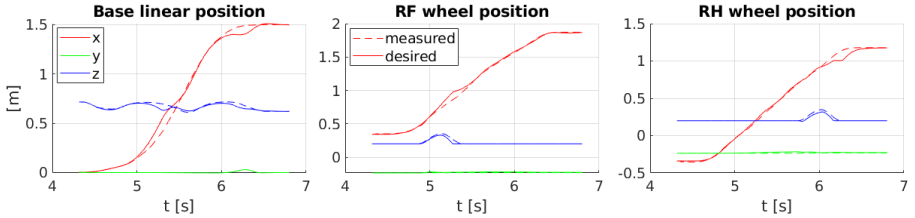

Fig. 8. Simulation results for the hybrid bounding maneuver over the gap. The desired positions provided as input to the WBC (dashed lines) compared with the simulated measured positions of the robot's base and wheels (full lines).

Fig. 9 shows the ANYmal robot traversing a floating step with a 0.2 m height (40% of the leg's length) at an average speed of 0.75 m/s, with a custom gait based on the hybrid gallop (Fig. 4(b)), adjusted to include a full stance phase after the swing of the RF wheel. Note how the robot lifts the front left wheel prior to the obstacle and it is already in the highest position when it reaches the step, which speeds up the maneuver. Even before the LF wheel reaches the step, the RF wheel starts to lift-off the ground. The same procedure is then carried out for the hind wheels to climb up the step. The average RMSE tracking error was less than 12 mm for this maneuver. To the best of our knowledge, such a fast negotiation of a step with a height of four times the wheel's radius (0.05 m) with a dynamic hybrid motion has never been shown before.

Next, the same step is placed on the right side of the robot's path, with the target position at a straight distance from the robot. A specific contact schedule was designed for such asymmetric obstacles, where the left wheels are in driving phase for the entire trajectory. The planned motion is presented in Fig. 10. As expected, the base finishes with a 10° roll angle to stay clear from the kinematic limits of the wheels. In general, this is an easier motion to execute, since only one wheel is lifted up at a time. This is translated to a RMSE tracking error of less the 3.0 mm for both the base and the wheels.

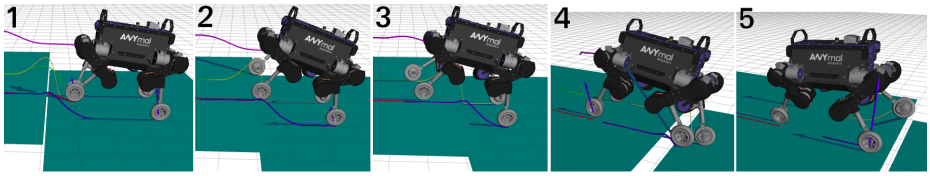

Fig. 9. The ANYmal robot performing a dynamic hybrid gallop gait to cross a step with a 0.2 m height.

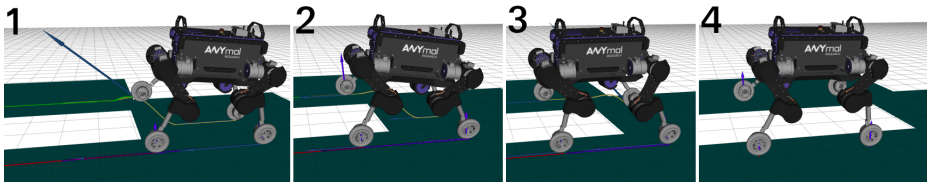

Fig. 10. The ANYmal robot crossing a 0.2 m high step on the right side of its path.

4 Conclusions

This paper presents a TO framework for wheeled-legged robots that is able to generate purely driving motions and simultaneous stepping and driving motions. Such approach takes full advantage of the hybrid nature of wheeled-legged robots and their high mobility in rough terrain. The proposed formulation takes into account the terrain information and the robot dynamics, which allows the robot to overcome challenging and discontinuous obstacles with dynamic motions. The optimized trajectories are verified in physical simulations with the ANYmal robot equipped with actuated wheels in different scenarios at speeds equal or higher than 0.75 m/s. Moreover, the approach can handle both statically and dynamically stable gaits, including gaits with full flight phases. Future work includes the implementation of a gait optimization feature in our formulation where the durations of each phase will become decision variables of the NLP. This would increase the number of motions that can be generated by the framework, but could cause a significant increase in the computational cost.

References

1. Bellegarda, G., Byl, K.: Trajectory optimization for a wheel-legged system for dynamic maneuvers that allow for wheel slip. In: 2019 IEEE Conference on Decision and Control (CDC) (2019)
2. Bjelonic, M., Bellicoso, C.D., de Viragh, Y., Sako, D., Tresoldi, F.D., Jenelten, F., Hutter, M.: Keep rollinwhole-body motion control and planning for wheeled quadrupedal robots. IEEE Robotics and Automation Letters **4**(2), 2116–2123 (2019)
3. Bjelonic, M., Sankar, P.K., Bellicoso, C.D., Vallery, H., Hutter, M.: Rolling in the deep hybrid locomotion for wheeled-legged robots using online trajectory optimization. IEEE Robotics and Automation Letters **5**(2), 3626–3633 (2020)

4. Cordes, F., Kirchner, F., Babu, A.: Design and field testing of a rover with an actively articulated suspension system in a mars analog terrain. Journal of Field Robotics **35**(7), 1149–1181 (2018)
5. de Viragh, Y., Bjelonic, M., Bellicoso, C.D., Jenelten, F., Hutter, M.: Trajectory optimization for wheeled-legged quadrupedal robots using linearized zmp constraints. IEEE Robotics and Automation Letters **4**(2), 1633–1640 (April 2019)
6. Fankhauser, P., Bloesch, M., Hutter, M.: Probabilistic terrain mapping for mobile robots with uncertain localization. IEEE Robotics and Automation Letters **3**(4), 3019–3026 (Oct 2018)
7. Geilinger, M., Poranne, R., Desai, R., Thomaszewski, B., Coros, S.: Skaterbots: Optimization-based design and motion synthesis for robotic creatures with legs and wheels. In: on Graphics (TOG), A.T. (ed.) Proceedings of ACM SIGGRAPH. vol. 37. ACM (August 2018)
8. Grand, C., Benamar, F., Plumet, F.: Motion kinematics analysis of wheeled–legged rover over 3d surface with posture adaptation. Mechanism and Machine Theory **45**(3), 477–495 (2010)
9. Hargraves, C., Paris, S.: Direct trajectory optimization using nonlinear programming and collocation. Journal of Guidance, Control, and Dynamics **10**(4), 338–342 (1987)
10. Jelavic, E., Hutter, M.: Whole-body motion planning for walking excavators. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2292–2299 (2019)
11. Klamt, T., Behnke, S.: Anytime hybrid driving-stepping locomotion planning. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4444–4451 (Sep 2017)
12. Klamt, T., Rodriguez, D., Schwarz, M., Lenz, C., Pavlichenko, D., Droeschel, D., Behnke, S.: Supervised autonomous locomotion and manipulation for disaster response with a centaur-like robot. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1–8 (Oct 2018)
13. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). vol. 3, pp. 2149–2154 (Sep 2004)
14. Medeiros, V.S., Jelavic, E., Bjelonic, M., Siegwart, R., Meggiolaro, M.A., Hutter, M.: Trajectory optimization for wheeled-legged quadrupedal robots driving in challenging terrain. IEEE Robotics and Automation Letters **5**(3), 4172–4179 (2020)
15. Reid, W., Prez-Grau, F.J., Gktoan, A.H., Sukkarieh, S.: Actively articulated suspension for a wheel-on-leg rover operating on a martian analog surface. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). pp. 5596–5602 (May 2016)
16. Smith, R.: Open dynamics engine (2008), `http://www.ode.org/`
17. Sun, J., You, Y., Zhao, X., Adiwahono, A.H., Chew, C.M.: Towards more possibilities: Motion planning and control for hybrid locomotion of wheeled-legged robots. IEEE Robotics and Automation Letters **5**(2), 3723–3730 (2020)
18. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming **106**(1), 25–57 (Mar 2006)
19. Winkler, A.W., Bellicoso, C.D., Hutter, M., Buchli, J.: Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. IEEE Robotics and Automation Letters **3**(3), 1560–1567 (July 2018)
20. Winkler, A.W.: Ifopt - A modern, light-weight, Eigen-based C++ interface to Nonlinear Programming solvers Ipopt and Snopt. (2018)