Trajectory Optimization for Wheeled Quadrupedal Robots Driving in Challenging Terrain

Trajectory Optimization for Wheeled Quadrupedal Robots Driving in Challenging Terrain

Vivian S. Medeiros[a], Marko Bjelonic[b], Edo Jelavic[b], Roland Siegwart[c], Marco A. Meggiolaro[a], Marco Hutter[b]
[a]Mechanical Engineering Department, PUC-Rio, Rio de Janeiro 22451-000, Brazil
[b]Robotic Systems Lab, ETH Zurich, 8092 Zurich, Switzerland
[c]Autonomous Systems Lab, ETH Zurich, 8092 Zurich, Switzerland

1 Introduction

Wheeled-legged robots are an attractive solution for autonomous locomotion in challenging terrain. They combine the efficiency of the wheels with the ability of the legs to traverse challenging terrain. Most of the research on wheeled-legged robots locomotion [1–3] focuses on designing a motion control framework that enables stable motion over uneven terrain, rather than generating optimized trajectories for dynamic motion. Recently, a motion control and planning framework for wheeled quadrupedal robots that continuously computes the reference trajectories for the robot's center of mass (CoM) using a Zero-Moment Point (ZMP) optimization has been presented in [4]. Other approaches involve driving over flat terrain and walking over irregular terrain [3,5]. No presented motion planner, however, considers the terrain information, and therefore, is not able to plan driving motions over larger obstacles. Negotiating larger obstacles is enabled through the presence of the legs and is not possible with a conventional wheeled robot.

We present a trajectory optimizer for driving motions for wheeled-legged quadrupedal robots with actuated wheels. A simplified two-dimensional Single Rigid Body model is used, which allows for fast solutions even for trajectories with a long time horizon. Since the planner has knowledge of the terrain and performs the optimization over the wheels' contact forces as well as its positions, the robot is able to traverse challenging terrain, including driving up a step, which to the best of our knowledge was never shown before. Fig. 1 shows the robot driving over a step.

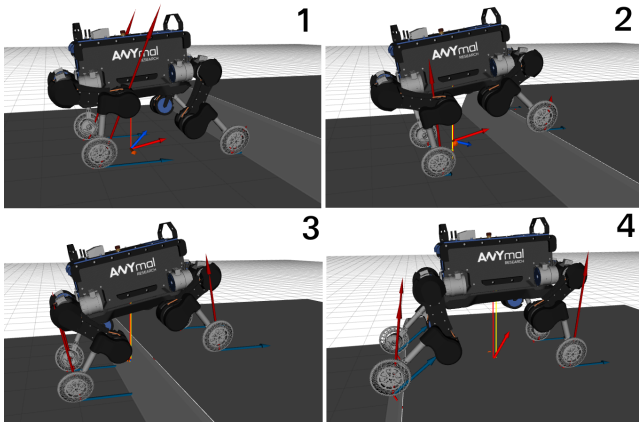

Figure 1: Simulation of the robot driving up a step whose height corresponds to 40% of the legs length using our trajectory optimization. Initially, the robot maintains its base slightly backward and moves the front wheels towards the step (1). Next, it moves the hind legs closer to the CoM to have higher traction with the hind wheels (2). This enables driving the front wheels up the step (3). Lastly, it moves the base closer to the ground and pulls the hind legs up (4), completing the maneuver. The dark red arrows on the wheels are the contact forces, the dark blue arrows are the velocities, and the smaller light red arrows are the accelerations of the wheels and the robot's CoM. A video demonstrating this maneuver can be found at https://youtu.be/lELr4stekhQ.

2 Trajectory Optimization

The robot's dynamic model used for trajectory optimization is based on a simplified two-dimensional Single Rigid Body Dynamics model. The legs' masses are assumed negligible compared to the base of the robot and a single rigid-body approximates the robot model with mass and inertia located at the robot's CoM. These assumptions are reasonable for most legged robots since the mass and the inertia of the legs are considerably less than the ones of the torso. Additionally, they make the dynamics of the robot independent of the joint configuration of the legs, thus keeping the formulation simpler and enabling fast convergence of the optimizer.

We formulate the motion planning as an Optimal Control Problem which we then transcribe into an Nonlinear Programming Problem (NLP) by dividing the trajectory into fixed time intervals ΔT, including the initial state, generating $n = \text{floor}(T/\Delta T) + 1$ nodes. All the optimization variables define each node at the time $t = k\Delta T$ of the trajectory. Once the problem is solved, the continuous solution can be obtained by linear interpolation between the nodes. The decision variables are the robot's CoM position and orientation, the CoM linear and angular velocities, the wheels' positions and velocities and the contact forces on each wheel. The 2D motions are transcribed into 3D by applying the

same trajectories to the left and right wheels and fixing the initial y-position of the base during the entire motion. The high-level user inputs are the final position of the CoM and the total time duration T of the trajectory.

Since this work focuses on the driving motions, all the wheels are constrained to remain in contact with the ground and the normal forces of the wheels must always be positive. The traction forces are limited according to the maximum torque of the wheel's motors and constrained to be inside the friction pyramid, to ensure that the no-slip condition is fulfilled. The dynamic consistency of the trajectory is ensured by enforcing the dynamic equations on the CoM accelerations. The main contribution of our planner is that it takes into account the stability of the robot along the whole trajectory. The stability criterion that we use is the Force-Angle Stability Measure presented in [6]. Lastly, kinematic consistency is enforced by limiting the distance between the wheels and the extension of the leg to stay in a reachable workspace that is known a priori and moves together with the robot. To reduce the computational cost of the optimization, we do not use a cost function, and the objective is to find the decision variables that fulfill all the constraints.

3 Results

The trajectory optimization is implemented in C++ using the Ifopt [7] interface for the interior-point method solver Ipopt [8]. The simulations are carried out in the robot simulation environment Gazebo with ODE [9] as the physics engine, using the full rigid body dynamics of the real quadrupedal robot equipped with actuated non-steerable wheels introduced in [4]. Besides, the torque limits of the actuators are considered in the simulation to ensure realistic results. The trajectories (base and wheels motions) are provided as input to the whole-body controller described in [4], that computes the actuation torques for the joints and the wheels.

The optimizer has been tested in different terrain types and the time interval between the nodes is chosen as 0.1s, which is short enough to ensure physically feasible and dynamically consistent motions. With this $\Delta T = 0.1$, and a 4.0s time horizon, the NLP has 720 optimization variables, 485 equality constraints and 813 inequality constraints. The solver computation time depends on the complexity of the terrain and remains on average 42ms per second of trajectory. Reported computational times have been obtained on a 2.7GHz dual-core Intel Core i7 laptop. Fig. 2 shows the desired motion computed by the trajectory optimizer along with the measured positions obtained during the simulation of the robot driving up a step. The whole-body controller is able to track the desired motions with Root-Mean-Square-Error (RMSE) less than 4cm. The maneuver is performed with an average speed of $0.5\text{m}/\text{s}$, which is as fast as walking up motions. The reader is encouraged to watch the accompanying video[1], that shows further motions in other terrains.

[1] Also available at https://youtu.be/lELr4stekhQ.

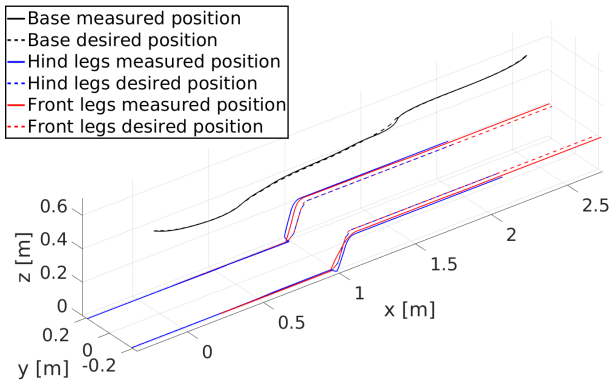

Figure 2: The desired positions provided as input to the whole-body controller and the simulated measured positions of the robot's base and wheels while driving up a step.

4 Conclusions

Our trajectory optimization framework is able to generate optimized motions for wheeled quadrupedal robots driving over challenging terrain by taking into account the terrain profile and the stability of the robot. The feasibility of the motion plans is demonstrated in simulations with ANYmal. In the future, experimental verification of these results will be carried out with the real robot. Furthermore, a new framework that uses a 3D model and allows for planning the walking motions is being developed. We also plan to enable the contact schedule optimization, similar to [10].

References

[1] Christopher Yee Wong, Korhan Turker, Inna Sharf, and Blake Beckman. *Posture Reconfiguration and Navigation Maneuvers on a Wheel-Legged Hydraulic Robot*, pages 215–228. Springer International Publishing, Cham, 2015.

[2] K. Turker, I. Sharf, and M. Trentini. Step negotiation with wheel traction: a strategy for a wheel-legged robot. In *2012 IEEE International Conference on Robotics and Automation*, pages 1168–1174, May 2012.

[3] T. Klamt, D. Rodriguez, M. Schwarz, C. Lenz, D. Pavlichenko, D. Droeschel, and S. Behnke. 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)*, pages 1–8, Oct 2018.

[4] 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*, pages 1–1, 2019.

[5] M. Schwarz, T. Rodehutskors, M. Schreiber, and S. Behnke. Hybrid driving-stepping locomotion with the wheeled-legged robot momaro. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5589–5595, May 2016.

[6] Evangelos Papadopoulos and Dariela Rey. The force-angle measure of tipover stability margin for mobile manipulators. *Vehicle System Dynamics - VEH SYST DYN*, 33:29–48, 01 2000.

[7] Alexander W Winkler. Ifopt - A modern, light-weight, Eigen-based C++ interface to Nonlinear Programming solvers Ipopt and Snopt., 2018.

[8] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, Mar 2006.

[9] Russell Smith. Open dynamics engine, 2008. http://www.ode.org/.

[10] 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*, 3(3):1560–1567, July 2018.