Trajectory Optimization for Legged Robots with Slipping Motions

Jan Carius[1], René Ranftl[2], Vladlen Koltun[2], and Marco Hutter[1]

***Abstract*—The dynamics of legged systems are characterized by under-actuation, instability, and contact state switching. We present a trajectory optimization method for generating physically consistent motions under these conditions. By integrating a custom solver for hard contact forces in the system dynamics model, the optimal control algorithm has the authority to freely transition between open, closed, and sliding contact states along the trajectory. Our method can discover stepping motions without a predefined contact schedule. Moreover, the optimizer makes use of slipping contacts if a no-slip condition is too restrictive for the task at hand. Additionally, we show that new behaviors like skating over slippery surfaces emerge automatically, which would not be possible with classical methods that assume stationary contact points. Experiments in simulation and on hardware confirm the physical consistency of the generated trajectories. Our solver achieves iteration rates of 40 Hz for a 1 s horizon and is therefore fast enough to run in a receding horizon setting.**

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

I. Introduction

LEGGED robots need to skillfully manipulate ground contact forces to propel themselves and retain stability. This article reports on our optimization approach for general articulated robots subject to contact interactions. By encapsulating a contact solving routine for hard unilateral contacts in the system dynamics, our motion planner can holistically optimize over whole-body motions and reason about contact sequence and timings simultaneously. Unlike most existing methods, we can also handle sliding contacts, a behavior that is traditionally avoided in motion planning algorithms.

Most established methods achieve advanced locomotion proficiency through a layered approach [1]–[3]. On the highest level, a navigation module generates a path that guides the robot's center to the goal while avoiding obstacles and impassable terrain. The middle layer is formed by a motion planner that often employs a simplified model of the robot dynamics and some stability criteria like the Zero-Moment Point [4] or Raibert heuristic [5]. The planner generates references on the joint or center of mass (COM) level and decides where each foot is placed on the ground and which forces shall be applied. To avoid tackling the combinatorial problem of contact selection, many approaches on this level assume a fixed contact schedule (e.g., manually pre-specified gait or given by an external contact planner [6], [7]). On the lowest level of locomotion control, a controller tracks the motion plan as close as possible and interacts with the actuators of the system. Engineered behaviors such as stepping reflexes and contact recovery are added at this level to improve robustness.

Fig. 1. The quadrupedal robot ANYmal tracking an optimized motion plan on a slippery surface. Contact forces are actively modulated to transition between stick and slip phases.

The community has embraced trajectory optimization (TO) as its dominant tool for the motion planner because physical constraints, goal specification, and a notion of quality and efficiency can be conveniently formalized in a single optimization problem. Unfortunately, the dynamics and constraints of walking machines lead to a difficult nonconvex optimization problem where discrete contact switching reduces the information content of gradients. Therefore, popular gradient-based sparse nonlinear program (NLP) solvers or differential dynamic programming (DDP) methods have difficulties when being initialized far from the optimal solution. Decoupling the problem into separate components alleviates some of the problems but may lead to infeasible motion plans or overly conservative behavior. On the other hand, solving the entire locomotion problem at once, i.e., a monolithic algorithm that computes physically correct joint references from a goal specification, is computationally very challenging [3], [8]–[11] and typically requires simplifying assumptions to produce solutions at interactive rates.

With this work, we present a solution to the motion planning problem that translates local goal specifications into feasible motion plans that capitalize on the full range of contact states.

Manuscript received: February, 24, 2019; Revised May, 13, 2019; Accepted June, 9, 2019.

This paper was recommended for publication by Editor Nancy Amato upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Intel Labs, the Swiss National Science Foundation (SNF) through project 166232 and the National Centre of Competence in Research Robotics (NCCR Robotics), and the European Union's Horizon 2020 research and innovation programme under grant agreement No 780883. This work has been conducted as part of ANYmal Research, a community to advance legged robotics.

[1]J.C. and M.H. are with the Robotic Systems Lab, ETH Zürich, Switzerland. jcarius@ethz.ch

[2]R.R. and V.K. are with the Intelligent Systems Lab, Intel.

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

While classically only a binary decision between open and closed contacts is made, we also allow for sliding contacts. Our objective is to design an accurate and efficient planner whose trajectories can be directly tracked without additional logic on the controller side. We show how a DDP-like planner can operate on a physically realistic model and generate consistent motions without a predefined contact (state) sequence while remaining computationally competitive. The use of a realistic contact model allows us to drop the no-slip assumption and thereby generate novel behaviors like walking with slipping contacts.

A. Related work

Many works in the legged motion planning domain focus on designing simplifications and proxy constraints to reduce the computational complexity and nonlinearity of the optimal control (OC) problem. Handling the full whole-body dynamics is often believed to be intractable without model reduction. A recent article [3] presents a state-of-the-art implementation of the layered approach introduced above and proposes numerous proxy constraints that appear well-behaved with the multiple shooting method and even generalize to non-flat ground.

Usage of centroidal dynamics is the most popular modeling choice in the literature [3], [10]–[12]. These methods allow planning over contact forces, contact locations, and COM motion in cartesian space. Since the effect of all forces is projected onto the COM frame, it is not clear yet how to allow active manipulation of internal angular momentum efficiently [13]. In contrast, we showcase an instance of TO with full dynamics in joint space, which allows us to precisely specify torque and velocity constraints of our actuators and generate motions that make use of the full dynamics of the system. While this induces a more complex optimization landscape, we show that we can still find motions at interactive solver rates.

Continuous OC problems for high-dimensional robotic systems are most commonly solved by direct methods. A transcription technique is thus required that turns the variational problem into a finite-dimensional mathematical program. Variants of DDP methods have proven to be a competitive option for transcription and were shown fast enough for receding horizon control [8], [12]–[14]. In this context, the required speed and convergence properties are frequently achieved by smoothing of physics constraints, soft contact models, or fixed gait specifications. Due to space constraints, we focus on the family of DDP methods in this review, although other options like direct collocation [9], [11], [15] or mixed integer programming [10], [16], [17] are also active fields of research.

Recently, we showed that contact-implicit optimization [18] can be achieved by solving hard contact constraints at the dynamics level through the inclusion of a time-stepping integration scheme. The encapsulating iterative linear-quadratic regulator (iLQR) [19] optimization procedure is able to discover hopping and walking motions for a single leg. In the present work, we extend this method and demonstrate that it scales to larger and more complex systems like a quadrupedal robot and thereby even finds previously unseen behaviors like contact sliding.

The motion optimization approach by Neunert et al. [14], [20] also eliminates the need for contact constraints by absorbing their effects into the state propagation law. With well-tuned cost function and contact parameters, their optimizer can discover feasible walking motions similar to our method. Our work can be seen as a generalization that replaces the soft contact model with a physically accurate contact solver that respects Coulomb's friction law by design. This results in much crisper contact interactions instead of "mud treading" motions and richer behavior depending on the configurable restitution and friction parameters.

B. Contributions

This work presents a complete pipeline of optimization-based motion planning for legged robots. We show how the full-body dynamics of a quadruped can be efficiently optimized under unilateral constraints with Coulomb friction, possible stick-slip transitions, and automatically emerging contact mode schedule. This article specifically reports the following results which we empirically validate on the quadrupedal robot ANYmal [21] (Fig. 1):

1) Our contact-implicit system model enables a single shooting method to discover diverse movement patterns of an underactuated and unstable walking robot.
2) Generated motions exploit sliding contacts and stick-slip transitions for both locomotion on slippery surfaces and to extend the degrees of freedom (DOFs) of the system, if necessary. To the best of our knowledge, this is the first instance of a motion planner that purposefully utilizes such contact interactions to create feasible walking trajectories.
3) The generated trajectories are physically accurate such that more than a dozen footsteps can be stably tracked without replanning.
4) The iterations of our solver are fast enough to be considered for use in a receding horizon implementation. We report on the disturbance rejection performance of our through-contact optimization scheme in a model predictive control (MPC) setting.

II. Method

Our strategy to generate motions for walking robots is to formulate and solve a suitable OC problem. In the following, we first describe how we model the state evolution of our system. Our state map comprises both the rigid body dynamics of the robot as well as the effects of contact constraints. Afterwards, the OC problem is formulated based on the system dynamics and subsequently solved.

A. Rigid body dynamics with unilateral constraints and friction

We model our system as an articulated rigid body subject to actuation torques (inputs $\boldsymbol{u}$) and N_c external forces $\boldsymbol{f}_{\text{ext}} = [\boldsymbol{f}_1^\top \cdots \boldsymbol{f}_{N_c}^\top]^\top \in \mathbb{R}^{3N_c}$ from the environment which arise from interactions with the ground. The contacts between the robot's feet and the ground plane are modeled as hard unilateral constraints with Coulomb friction and zero restitution.

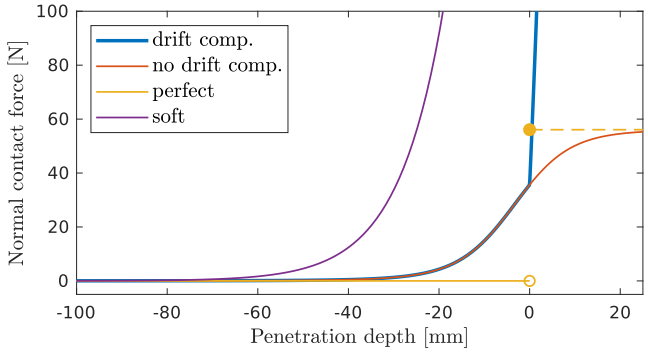

Fig. 2. Comparison of contact models as a function of penetration at zero velocity. The soft contact [14] is not aware of the inertia at the foot; hence contact penetration becomes a function of the load. Drift compensation aggressively increases the repulsion force as penetration occurs but does not affect restitution behavior. Without penetration, the drift and no-drift models coincide. The "perfect" contact force is given as a reference and corresponds to the quasi-static forces required for compensating gravity. Penetration should never occur in the perfect case (dashed line).

Algorithm 1 State integration by time-stepping

Input: Initial state $\boldsymbol{x}_0$ and zero-order hold control input $\boldsymbol{u}$
Forward-Euler half step for joint positions and base pose:
- $\boldsymbol{x}_m \leftarrow \boldsymbol{x}_0 + \texttt{forwardEulerPositionUpdate}(\boldsymbol{x}_0, \Delta t/2)$
Calculate quantities for contact solver:
- $\boldsymbol{d} \leftarrow \texttt{calculateFeetToGroundDistances}(\boldsymbol{x}_m)$
- $\boldsymbol{J_c} \leftarrow \texttt{stackedFootJacobians}(\boldsymbol{x}_m)$
- $\boldsymbol{G} \leftarrow \boldsymbol{J_c}\ \texttt{massMatrix}^{-1}(\boldsymbol{x_m})\ \boldsymbol{J_c}^\top$
- $\tilde{\boldsymbol{G}} \leftarrow \boldsymbol{G} + \texttt{complementarityCorrection}(\boldsymbol{d})$
- $\boldsymbol{c} \leftarrow \texttt{collectEomTerms}(\boldsymbol{x}_m, \Delta t)$
- $\tilde{\boldsymbol{c}} \leftarrow \boldsymbol{c} + \texttt{driftCompensation}(\boldsymbol{d}, \Delta t)$
Solve for contact forces:
$\boldsymbol{f}^{\text{ext}} \leftarrow \texttt{solveQuadraticProgram}(\tilde{\boldsymbol{G}}, \tilde{\boldsymbol{c}})$
Velocity and position update step:
- $\boldsymbol{x}_e \leftarrow \boldsymbol{x}_0 + \texttt{forwardEulerVelocityUpdate}(\boldsymbol{x}_0, \boldsymbol{f}^{\text{ext}}, \Delta t)$
- $\boldsymbol{x}_e \leftarrow \texttt{trapezoidalPositionUpdate}(\boldsymbol{x}_0, \boldsymbol{x}_e, \Delta t)$
Output: State $\boldsymbol{x}_e$ after time step

The system's state $\boldsymbol{x} \in SE(3) \times \mathbb{R}^{6+2N_j}$ comprises the robot's position, orientation, and twist as well as the position and velocity of all N_j joints. We use exponential map parameters to represent 3D orientation because it has a minimal representation and no singularities in the range of motion that we are interested in. Both $\boldsymbol{x}[n]$ and $\boldsymbol{u}[n]$ are functions of the time step n.

Alg. 1 summarizes the key elements to obtain the discrete-time state propagation law $\boldsymbol{x}[n+1] = \boldsymbol{f}(\boldsymbol{x}[n], \boldsymbol{u}[n])$ by a time-stepping scheme [18]. The generalized positions are first updated by a forward-Euler step over half the time interval. Then we calculate the contact forces by solving a quadratic program with second-order cone constraints:

$$\underset{\{\boldsymbol{f}^{\text{ext}} \in \mathcal{F}\}}{\text{minimize}} \quad \frac{1}{2} \boldsymbol{f}^{\text{ext}\top} \boldsymbol{G} \boldsymbol{f}^{\text{ext}} + \boldsymbol{f}^{\text{ext}\top} \boldsymbol{c} \,. \tag{1}$$

The positive-definite Delassus matrix $\boldsymbol{G} \in \mathbb{R}^{3N_c \times 3N_c}$ corresponds to the inverse inertia felt at the contact points. The vector $\boldsymbol{c} \in \mathbb{R}^{3N_c}$ collects other equations of motion (EOM) terms such that the optimization (1) finds the minimal feasible contact forces that reduce the contact point velocities to zero in the next time step (principle of least action). The feasible set $\mathcal{F}$ models Coulomb friction of the unilateral contacts. Once the contact forces are obtained, the velocities at the end of the integration interval are calculated by a forward-Euler step using the generalized accelerations (i.e., forward dynamics). Last, the final generalized positions are given by the implicit trapezoidal integration rule on the velocities.

Thus far, we have ignored the fact that forces only act when the separation between a point on the robot and the environment is zero. In our previous work, we selectively added only those contact forces $\boldsymbol{f}_i$ to the stacked vector $\boldsymbol{f}_{\text{ext}}$ whose separation distance d_i was nonpositive. In the present work, we consider contacts to be always active. This modification avoids changing the dimensions of the optimization problem (1) and ensures the forces remain a continuous function of the state. In order to still comply with the complementarity condition between reaction force and separation distance d_i, we compel the normal component of the reaction force to zero by adding a large exponential term on the corresponding diagonal entry of the $\boldsymbol{G}$ matrix [22]

$$\hat{\boldsymbol{G}}_{ii} = \boldsymbol{G}_{ii} + \exp\left(c_1 \tanh\left(c_2\, d_i\right)\right)/c_3 \,. \tag{2}$$

The coefficients $c_1 = 8, c_2 = 20, c_3 = 10$ are chosen such that all relevant effects like restitution and friction are still captured while contact forces are reduced very quickly with increasing distance. In particular, it ensures that the contacts appear very crisp, surpassing a soft contact model.

A further extension to [18] is the addition of drift stabilization according to the modified Θ-method [23] to prevent the feet from sinking into the ground during a stance phase. Drift into the constraint surface is possible because our time-stepping method enforces the unilateral constraint only on the velocity level. Constraint satisfaction on position level is recovered by adding a compensation term $d_i/\Delta t$ to the row of $\boldsymbol{c}$ that corresponds to the violating contact i. The additional contribution acts like a predictive element that induces a force that pushes the contact point out of the constraint. We find that this seemingly small modification helps the solver to find a better motion because contact opening is more straightforward to discover when the feet stay on the contact surface. The behavior of our hard contact model is compared to alternative formulations in Fig. 2.

In the following optimization, the partial derivatives of the state map $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}, \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}$ will be required. We obtain these state and input dependent Jacobian matrices through automatic differentiation (AD). The evaluation of the function $\boldsymbol{f}$ is recorded in a tape of operations that reproduces Alg. 1 including all underlying rigid body dynamics algorithms, the contact solving loop, and the time-stepping integration scheme. As a consequence, the partial derivatives of this routine can accumulate enormous length on the order of 100'000 generated lines of code. While this is not a problem for the AD suite (CppAD), the compilation of this code can become problematic. We reduce the complexity of evaluating the derivative by omitting the drift compensation from the tape. Furthermore, we eliminate unnecessary operations by exploiting the symmetry of inertia tensors.

B. The optimal control problem

We consider a discrete, finite horizon optimal control problem of the form

$$\underset{\boldsymbol{u}[\cdot]}{\text{minimize}} \quad J = \Phi(\boldsymbol{x}[n_f]) + \sum_{n=0}^{n_f-1} l(\boldsymbol{x}[n], \boldsymbol{u}[n], n) \ , \tag{3}$$

$$\text{subject to} \quad \begin{cases} \boldsymbol{x}[n+1] = & \boldsymbol{f}(\boldsymbol{x}[n], \boldsymbol{u}[n]) \ , \\ \boldsymbol{x}[0] = & \boldsymbol{x}_0 \ , \\ \boldsymbol{S}\,\boldsymbol{x}[i] \leq & \boldsymbol{v}_{\text{jointLimits}} \qquad \forall i \ , \\ ||\boldsymbol{u}[i]||_\infty \leq & u_{\max} \qquad \forall i \ , \end{cases} \tag{4}$$

where n_f is the number of time steps and $\boldsymbol{x}_0$ a given initial state. We use quadratic final costs of the form

$$\Phi(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_{\text{ref},f})^\top \boldsymbol{R}_f (\boldsymbol{x} - \boldsymbol{x}_{\text{ref},f}) \ , \tag{5}$$

and running costs of the form

$$\begin{aligned} l(\boldsymbol{x}, \boldsymbol{u}, n) = (\boldsymbol{x} - \boldsymbol{x}_{\text{nom}})^\top \boldsymbol{Q} (\boldsymbol{x} - \boldsymbol{x}_{\text{nom}}) + \boldsymbol{u}^\top \boldsymbol{R} \boldsymbol{u} \\ + a[n] (\boldsymbol{x} - \boldsymbol{x}_{\text{ref}}[n])^\top \boldsymbol{Q}_a (\boldsymbol{x} - \boldsymbol{x}_{\text{ref}}[n]) \ . \end{aligned} \tag{6}$$

The scalar $a[n]$ is a time-dependent cost activation switch that we use to encourage time-encoded behaviors like periodic walking. The nominal configuration $\boldsymbol{x}_{\text{nom}}$ acts as a regularizer while reference configurations $\boldsymbol{x}_{\text{ref}}$ encode specific goals such as moving individual limbs or the robot's torso. Our dynamics $\boldsymbol{f}$ already include the effects of contact forces, so no further contact constraints are required in (4). Additional constraints, e.g., to enforce joint limits, can be added to the OC problem through an appropriate selection matrix $\boldsymbol{S}$, although we find that the existence of regularization costs makes them needless.

Our solution strategy is closely related to the well-known iLQR and can be seen as a direct single shooting approach. In each iteration of the algorithm, we roll out the current control trajectory $\boldsymbol{u}[\cdot]$ under the nominal dynamics $\boldsymbol{f}$, resulting in the nominal state trajectory $\boldsymbol{x}[\cdot]$. Subsequently, a local linear approximation of the dynamics and a quadratic approximation of the cost are calculated by automatic differentiation. The resulting linear-quadratic optimal control (LQOC) problem is then passed to the HPIPM solver [24] to obtain nominal state and control updates for the next iteration together with a linear feedback law. Finally, we perform a backtracking line search on the update size to ensure decreasing costs.

The advantage of using a shooting method with our model $\boldsymbol{f}$ is that all motions are guaranteed to remain feasible and the optimizer is relieved from handling the intricate complementarity constraints that arise from contact interactions. Furthermore, the optimizer can reason about all modeled behaviors such as stick-slip transitions and making or breaking contact at any time.

To enforce input limits $u_{\max}$ we obtain similar results by either enforcing the constraints on the solver level or re-parameterizing the input through a sigmoid transform

$$\hat{\boldsymbol{u}}_i = u_{\max} \tanh\left(\frac{2\boldsymbol{u}_i}{u_{\max}}\right) \ , \tag{7}$$

where only the transformed input $\hat{\boldsymbol{u}}$ is applied to the actual dynamics. In practice, we use the transform method to gain a small speed-up in the solver. Additional state or state-input constraints can also be passed to the HPIPM solver.

C. Application on a robot

Solving the OC problem (3), (4) results in discretized state and control trajectories. To track the motion plan on the robot, we linearly interpolate these trajectories at the required control frequency to obtain the planned state and input at the current time. The tracking controller then applies the optimized torques to achieve the planned accelerations and contact forces and closes a PD feedback loop on joint positions and velocities. Relatively stiff joint tracking gains are necessary to control motion plans with slipping contacts such that stick-slip transitions are followed even if the encountered friction is higher or lower than initially planned for.

While more sophisticated control architectures are available, e.g., hierarchical optimization-based whole-body control, such controllers usually require specifying which leg can be used as support and which is in swing phase. Our contact-invariant motion plans do not make a clear distinction between these phases and may additionally include sliding contacts. The trajectories we produce are accurate enough such that no advanced stabilizing action from the controller is required. Therefore, we apply the described simple tracking controller and show that it is sufficient for realizing the trajectories on hardware.

III. Results

We present quantitative and qualitative results on the performance of our algorithm. Moreover, we demonstrate the validity of the resulting motion plans through experiments in a physics simulation environment (Gazebo with Open Dynamics Engine (ODE)) and on real hardware.

A. Test setup

We evaluate our algorithm on the quadrupedal robot ANYmal [21] shown in Fig. 1. It has $N_j = 12$ torque-controlled joints and $N_c = 4$ point-feet on which contact forces are acting. For motions on regular ground, we set a friction coefficient of 0.7 in the contact solver. The dynamics model parameters[1] have to be tuned only once for a given system and environment. We choose a time step of $\Delta t = 0.01\,\text{s}$. The regularization cost terms can be kept constant across all experiments. For task-specific costs, our procedure is to gradually increase their weights until the desired behavior can be observed. All motion plan optimizations are initialized with a trivial standing position at nominal joint angles.

B. Discovery of walking motion and gait sequence

Contrary to switched dynamic models, our optimizer has the authority to manipulate the contact state at each time step because we model the system dynamics *including* the effects of contact constraints. In this subsection, we thus evaluate our algorithm's ability to exploit these additional DOFs and automatically discover a valid walking motion with multiple steps. In our initial experiments, we nudge the optimizer towards a specific contact sequence via modifications to the cost

[1]Contact properties, contact solver iteration settings, timestep

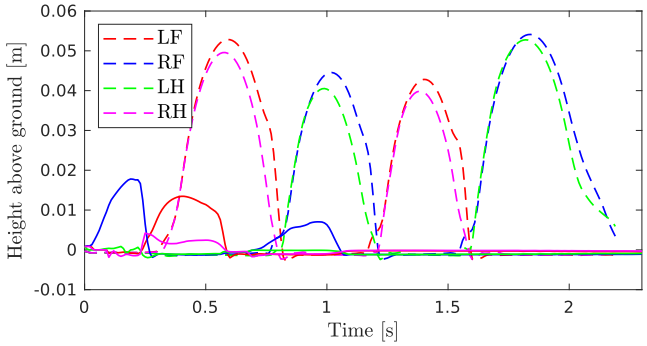

Fig. 3. Visualization of the foot (Left Front to Right Hind) positions from an optimized trajectory with hinted gait (dashed) and automatically emerging gait (solid, same colors). The motion without specified footfall pattern is less smooth and exhibits foot scuffing in the second half.

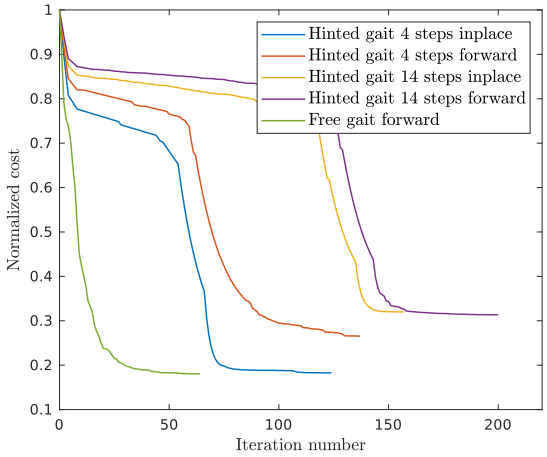

Fig. 4. Progression of the iterative optimization algorithm. All costs are normalized by their initial value for comparability. The free gait motion has a similar duration as the fixed gait motion with four steps. The sharp descent for motions with hinted gait happens when the algorithm discovers that contacts can be broken.

function. In the second set of experiments, the cost function only encodes that it is desirable to move forward, but no information about a favored leg motion is provided.

1) Motion plans with hinted gait: We first assess how the proposed optimization method performs with an unstable and underactuated system under switching contact modes. To encourage a multi-step motion, we encode a periodic trotting gait in the time-dependent portion of the running cost (6) by alternatingly setting the reference for two diagonal legs to a bent configuration. Such encoding of the gait was also previously used in [14].

With these cost modifications in place, we optimize trajectories of various lengths and both with and without forward motion. The resulting motions feature a smooth and visually appealing trotting gait as can be seen from the foot trajectories plotted in Fig. 3 and in the accompanying video[2]. The convergence behavior in Fig. 4 shows that a large number of iterations is initially required to discover that contacts can be opened. In this phase, the optimizer attempts to gradually unload the legs that are requested to be lifted and eventually discovers that contacts can be broken. Once the contact sequence is determined, the cost decreases rapidly. That means, if the initialization had been a stepping motion, the algorithm would have converged much faster. However, finding such a stable input sequence for the initialization is difficult in practice, and we evaluate our algorithm on its ability to find a walking motion from scratch.

2) Motion plans with emerging gait: Having verified that our algorithm can optimize across contact events, we now show that a feasible stepping motion can be discovered without any hints that the feet can or should be lifted. We optimize a trajectory of 2 s length (similar duration to the 4-step sequence with hinted gait) where we encourage a 0.5 m forward displacement of the base[3] but do not use any time-dependent costs that would induce a periodic gait.

We find that the balance between forward pushing costs and regularization terms is the dominating factor that influences the result of the optimization. Too little weight on the displacement cost makes the robot merely lean forward as far as the support polygon permits, but it does not discover a step. Overly strong terms can cause aggressive forward leaps that will not be robust in reality and bring the solver into an area of the state space that is difficult to navigate while keeping stability in successive updates. Our trials showed that the ratio between regularization costs and goal-directed costs should be approximately 1:1 in the initialization. In that case, a moderate forward motion emerges that partially resembles a trotting gait. The corresponding trajectories are plotted in Figs. 3 and 4, and a visualization is shown in the video. As we expect, the footfall pattern is not as clean as with a hinted gait, and the movement appears more impulsive. Still, the entire trajectory is within the physical constraints of our model, and the robot achieves to move forward without any human-designed gait specification. The solver converges to a trajectory that reaches the desired base position within 3% relative error.

The hardest challenge in achieving sensible motions without a constrained gait pattern is posed by the trade-off between local minima and scuffing feet. Our dynamics model only considers flat ground, so there is little reason for high ground clearance in a walking motion. The optimizer will, therefore, converge to trajectories that barely lift the feet, which is less robust in real-world scenarios. We establish that the drift compensation (Sect. II-A) facilitates proper contact opening significantly, whereas introducing additional nonlinearities in the cost function did not help reliably. The trajectories that we can currently produce exhibit foot scuffing to some degree but, as we will show later, this does not hinder hardware transfer and therefore confirms that the optimizer is operating within the physical limits of the contact behavior.

C. Motions with slippage

Given a variety of cost functions, the proposed algorithm can discover qualitatively different behaviors that exploit the contact model in multiple ways. For example, if we want the robot to turn in place and torques are very cheap, the resulting motion will be a jump that brings the robot fast and very effectively to the required yaw angle. As we increase the cost

[2] `https://youtu.be/Sd6lqoj9Tvs`

[3] Without a cost component that encourages forward motion, the algorithm converges to a stable standing position close to the initialization. Stepping in place cannot be achieved because it would always be suboptimal to lift a foot.

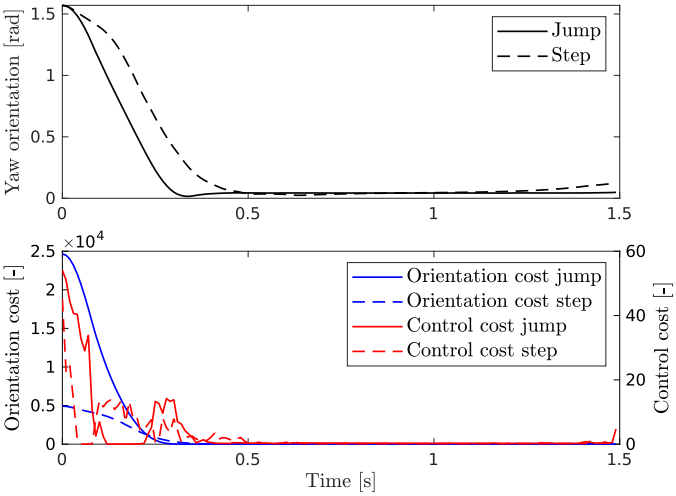

Fig. 5. Different cost weightings on yaw error result in qualitatively different emerging behaviors for 90° in-place rotations. A strong orientation penalty justifies larger cost expenditure for inputs and leads to a single jump instead of an energy-saving stepping motion.

of torques, the algorithm converges to a stepping motion that is less aggressive but drags the feet along the ground. Fig. 5 displays the corresponding plots.

In this section, we focus on our solver's ability to produce motions that explicitly make use of slipping contacts. Such movements would not be possible with TO algorithms that assume a no-slip condition for closed contacts.

1) Enabling motions that require slippage: We first consider a scenario where slippage is required to achieve a given task. To this end, we lock the hip abduction adduction (HAA) joints of our quadruped and impose costs to make the robot move along a circular arc (i.e., diagonally forward and simultaneously yaw with the torso). This task can only be achieved when one or more feet slip along the ground, similar to the turning motion of a skid-steer vehicle. Hence, only the active use of sliding contacts allow the system to regain the lost DOFs that are necessary to reach the goal.

Our algorithm can solve this task by selectively unloading feet that are supposed to move along the floor and pushing them across the stiction limit. Specific knowledge and authority over the Coulomb friction behavior allow the optimizer to find contact force distributions that make partially unloaded feet slide laterally on the ground while keeping the main body balanced. Inspired by the circular arc that is described by the robot's center during this trajectory, we find that tracking this motion plan repeatedly (i.e., looping the trajectory) results in walking a full circle. Fig. 6 shows how the robot progresses by repeatedly following the same motion reference.

2) Walking on slippery ground: Another interesting application for sliding contacts is walking on surfaces with a low friction coefficient. We create such a condition by placing our quadruped on a wet whiteboard. We determine the friction coefficient between the feet of our quadruped and the slippery whiteboard to be approximately $\mu = 0.3$ and assign this value to the corresponding contact model parameter.

Optimizing a motion with such low friction and without gait specification yields a "skating" behavior where alternatingly the loaded legs push the unloaded ones across the ground. We

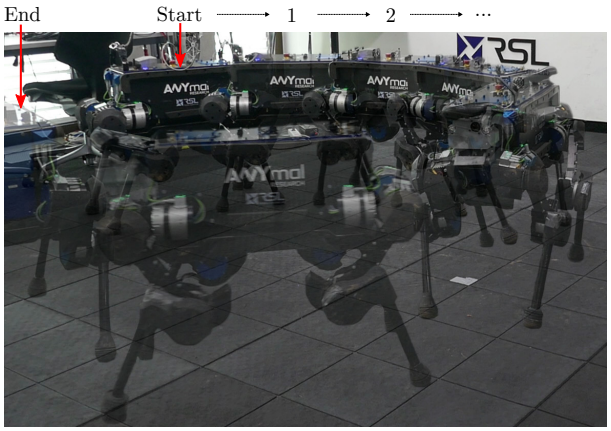

Fig. 6. Slippage enables curve walking even when the hip abduction DOFs are locked. The motion plan only spans two seconds, but repeated execution results in tracing out an entire circle.

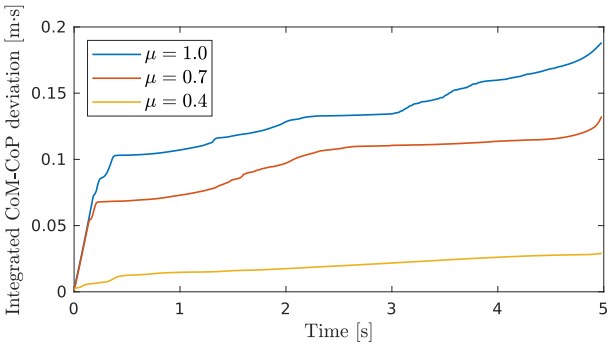

Fig. 7. Integrated deviation between COM and center of pressure (COP) along a trajectory with forward displacement for different friction coefficients. A larger deviation implies more aggressive accelerations. As friction decreases, the COP remains more closely below the COM.

observe that, compared to regular ground, a higher penalty on joint velocity is required to avoid oscillations in the torque trajectories due to frequent stick-slip transitions.

We find that confronting our algorithm with a low friction scenario allows it to embrace sliding as the preferred means of locomotion. This strategy has the potential to increase robustness because the feet are always kept close to the ground and can catch an unexpected slip of a support leg much quicker. We show in Fig. 7 that the planner distributes the contact forces more evenly as the friction coefficient decreases. This trend agrees with the natural behavior of humans to center their COM over the supporting limbs as the floor becomes slippery [25].

D. Physical accuracy of motion plans: hardware transfer

We apply the generated motions without further refinement on the physical system and use the simple tracking controller explained in Sec. II-C. The control frequency on our system is 400 Hz. No corrective behavior for early or late contact switching is in place. The fact that we can execute various motion plans on the hardware without replanning, in particular, the 14-step trotting sequence, confirms qualitatively that the generated trajectories are physically consistent.

For a more objective assessment, we compare the trajectories with hinted gait (4 steps, Sec. III-B1), emerging gait (Sec. III-B2), turning with reduced DOFs (Sec. III-C1), and skating on slippery surfaces (Sec. III-C2), each in simulation and when executed on hardware.

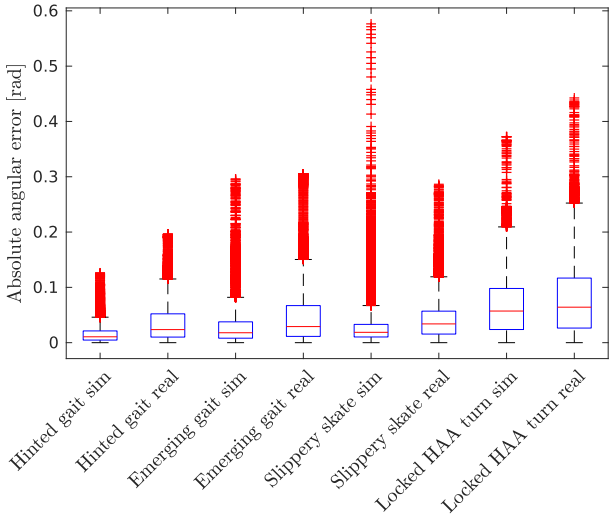

Fig. 8. Summary statistics of absolute joint tracking errors across all joints and for the entire trajectory. We show box plots for both motions with hinted and emerging gait as well as for trajectories that include slippage. Tracking performance is better in simulation (sim) than on real hardware. Slipping motions are harder to track on average, and there are more severe outliers.

TABLE I
TASK-SPECIFIC ERROR BETWEEN PLANNED AND EXECUTED MOTION

Relative error	Hinted gait	Emerging gait	Slippery skate	Locked HAA turn
Sim	4.82%	20.4%	15.5%	4.23%
Real	8.43%	38.0%	36.6%	34.8%

We first assess tracking performance at the joint level. Fig. 8 shows that the difficult motions have a mean angular error of up to $0.06\,\mathrm{rad}$ while the easier ones stay well below that. A general trend is that motions with more slippage lead to a more substantial mean tracking error as well as to more severe outliers. Errors are accumulated when a contact point unexpectedly starts slipping or, conversely, does not begin to slip although intended.

The deviation between plan and execution is consistently lower in the simulation. This result is expected due to model mismatch, different friction behavior, additional disturbances, and imperfect state estimation on the real system. In the case of skating on slippery ground in simulation, most of the severe outliers are collected within the initial movement phase because the robot's legs slide into a splayed configuration before we activate our controller. This offset explains why the outliers appear worse in simulation than on the real system.

Task-specific performance measures are shown in Tab. I. We compare the experimentally obtained forward displacement (hinted and emerging gait, slippery skate) or angle turned (locked HAA turn) with the optimized values. We observe that the pose of the robot begins to deviate as a result of unaccounted slippage effects, in particular on hardware. We conjecture that Coulomb's friction model proves too simplistic as the motion plan increasingly exploits sliding contacts.

Finally, we assess the accuracy of our motion plan in terms of the predicted contact forces. To this end, we attach a force sensor to the right front foot of our quadruped and plot its readings in Fig. 9 against the nominal plan. We conclude that the contact solver calculates ground reaction forces that closely

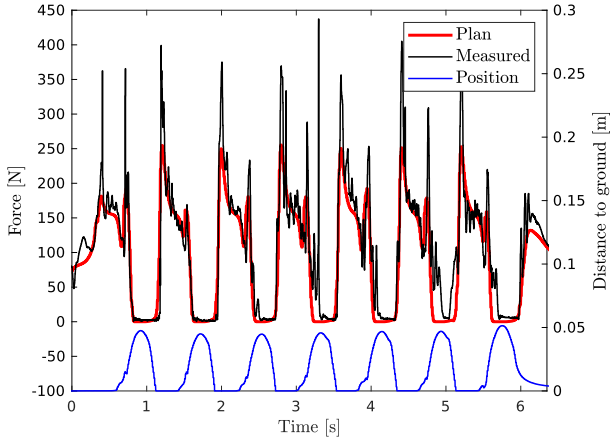

Fig. 9. Contact force tracking on hardware for a single foot during the 14-step forward motion. The plot compares planned with measured force magnitude and shows the foot position underneath to indicate contact phases.

match reality. Note that the spikes on touchdown are expected to be higher for the force sensor because the time stepping algorithm computes an average force over the integration step and cannot resolve such short impulses.

E. Replanning with receding horizon

Our motion planning algorithm is executed on a standard desktop computer with an Intel i7-8700K (3.7 GHz) CPU. By retaining the structure of the OC problem, we attain a computational complexity that is linear in the time horizon of the problem. On our machine, we achieve a solver iteration frequency of $39.3\,\mathrm{Hz}$ (std. dev. $1.29\,\mathrm{Hz}$) for a one-second horizon. While objective comparisons are difficult due to different hardware and optimization levels, our method can be considered on par with the fastest known methods for computing full-body motions [3, Sec. VI-B].

The speed of our algorithm allows us to run the planner in a receding horizon mode where we execute one solver iteration (rollout, backward pass, and line search) per control update. We find that the drift compensation from Sec. II-A leads to a higher rate of unstable rollouts because the measured state may put the robot's feet into the ground, leading to large repulsion forces. As the robot's state is frequently reset, we disable the drift compensation in MPC mode.

Continuous replanning should lead to improved disturbance rejection because the planner receives feedback about state deviations and can react accordingly. We test this conjecture by imposing a single quadratic objective that penalizes deviations from the nominal state. Without disturbance, MPC results in stable standing. Fig. 10 shows the result of applying an external lateral force to the main body. The force pulls the body to the left side and causes a deviation from the reference position. In order to counter the external force, the motion plan requests stronger contact forces on the left legs. Once the force is released, the base returns to the desired zero position smoothly and without overshoot.

Thanks to our through-contact optimization scheme, the controller keeps the system in balance even when the disturbance is strong enough to make a foot lose contact unexpectedly. We can qualitatively reproduce the same result on hardware as shown in the video.

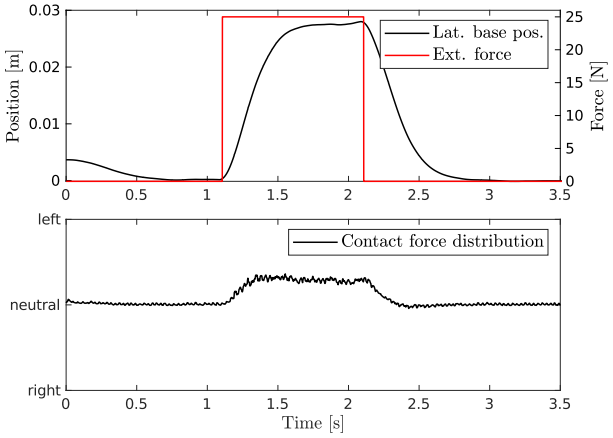

Fig. 10. Disturbance rejection under MPC in simulation. The top plot shows the lateral base position when subjected to a sideways force for one second. Underneath we visualize how the vertical contact force distribution on the feet shifts to the left to counter the external force and afterwards returns to a uniform distribution.

IV. Conclusion

We presented a motion planning algorithm for legged robots that can exploit sliding contacts. The physical feasibility of the generated trajectories was verified through multiple experiments in a physics simulator and on hardware. We showed that overcoming the typical no-slip constraints in motion planning creates emerging forms of locomotion that were not elicited before by comparable approaches. The additional DOFs gained by involving slippage in the motion plan makes the proposed type of motion optimization particularly applicable to walking robots with a limited number of joints.

Despite our results, the discovery of contact switches remains the hardest part in contact-implicit motion optimization due to limited gradient information. Possible remedies may include continuation approaches or sampling-based methods.

We have shown the potential of our algorithm in an MPC setting. In the future, we want to also demonstrate walking motions with continuous replanning, which is currently still fragile due to contact state mismatches between the real system and nominal model. For a robust deployment, state estimation and motion planning may need to interact more closely such that slipping feet can be anticipated in the estimator and accurate state updates are fed back to the planner. Our method is also not restricted to Coulomb's friction law, the predictive power of which is known to degrade as slippage occurs. We expect that online estimates of the contact properties would allow the MPC to handle the unpredictability of contact better.

One interesting future research direction is the application of our ideas in the field of dynamic manipulation. Tasks like reorienting an object in a (robotic) hand are good examples where sliding contacts are the natural way of interaction that make manipulation robust against unknown shape and mass distribution of the manipulated object.

References

[1] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Hierarchical planning of dynamic movements without scheduled contact sequences," in *Proc. Int. Conf. Robot. Autom*, 2016, pp. 4636–4641.

[2] J. Carpentier, A. Del Prete, S. Tonneau, T. Flayols, F. Forget, A. Mifsud, K. Giraud, D. Atchuthan, P. Fernbach, R. Budhiraja, *et al.*, "Multi-contact locomotion of legged robots in complex environments–the loco3d project," in *RSS Workshop on Challenges in Dynamic Legged Locomotion*, 2017.

[3] J. Carpentier and N. Mansard, "Multicontact locomotion of legged robots," *IEEE Trans. Robotics*, vol. 34, no. 6, pp. 1441–1460, 2018.

[4] M. Vukobratovic and B. Borovac, "Zero-moment point - thirty five years of its life," *I. J. Humanoid Robot.*, vol. 1, no. 1, pp. 157–173, 2004.

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

[6] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Proc. Humanoids*, 2014, pp. 279–286.

[7] S. Tonneau, A. D. Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots," *IEEE Trans. Robotics*, vol. 34, no. 3, pp. 586–601, 2018.

[8] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *Proc. Int. Conf. Intell. Robots Syst.*, 2012, pp. 4906–4913.

[9] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *I. J. Robotics Res.*, vol. 33, no. 1, pp. 69–81, 2014.

[10] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernández-López, and C. Semini, "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2531–2538, 2018.

[11] 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.

[12] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An efficient optimal planning and control framework for quadrupedal locomotion," in *Proc. Int. Conf. Robot. Autom*, 2017, pp. 93–100.

[13] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "Differential dynamic programming for multi-phase rigid contact dynamics," in *Proc. Humanoids*, 2018.

[14] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[15] D. Pardo, M. Neunert, A. W. Winkler, R. Grandia, and J. Buchli, "Hybrid direct collocation and control in the constraint-consistent subspace for dynamic legged robot locomotion," in *Robotics: Science and Systems XIII*, 2017.

[16] K. Wampler and Z. Popovic, "Optimal gait and form for animal locomotion," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 60:1–60:8, 2009.

[17] T. Marcucci, R. Deits, M. Gabiccini, A. Bicchi, and R. Tedrake, "Approximate hybrid model predictive control for multi-contact push recovery in complex environments," in *Proc. Humanoids*, 2017, pp. 31–38.

[18] J. Carius, R. Ranftl, V. Koltun, and M. Hutter, "Trajectory optimization with implicit hard contacts," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3316–3323, 2018.

[19] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference*, 2005, pp. 300–306.

[20] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory optimization through contacts and automatic gait discovery for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1502–1509, 2017.

[21] M. Hutter, C. Gehring, A. Lauber, F. Günther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Blösch, H. Kolvenbach, M. Bjelonic, L. Isler, and K. Meyer, "Anymal - toward legged robots for harsh environments," *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, 2017.

[22] T. Erez and E. Todorov, "Trajectory optimization for domains with contacts using inverse dynamics," in *Proc. Int. Conf. Intell. Robots Syst.*, 2012, pp. 4914–4919.

[23] M. Jean, "The non-smooth contact dynamics method," *Comput. Methods in Appl. Mechanics and Eng.*, vol. 177(3-4), pp. 235–257, 1999.

[24] G. Frison, H. H. B. Sorensen, B. Dammann, and J. B. Jørgensen, "High-performance small-scale solvers for linear model predictive control," in *European Control Conference (ECC)*, 2014, pp. 128–133.

[25] G. Cappellini, Y. P. Ivanenko, N. Dominici, R. E. Poppele, and F. Lacquaniti, "Motor patterns during walking on a slippery walkway," *Journal of Neurophysiology*, vol. 103, no. 2, pp. 746–760, 2010.