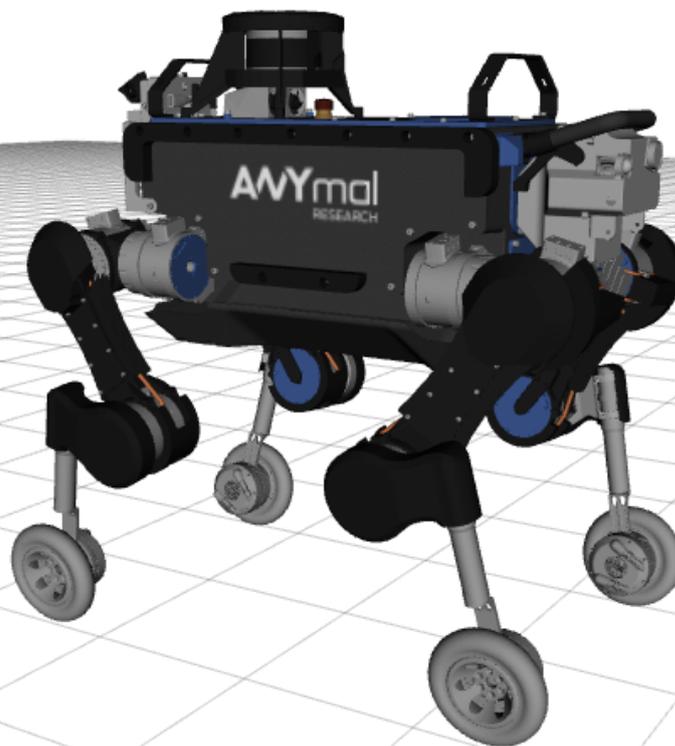


Trajectory Optimization for Hybrid Walking-Driving Motions on Wheeled Quadrupedal Robots

Prajish Kumar Sekoor Lakshmana Sankar



Trajectory Optimization for Hybrid Walking-Driving Motions on Wheeled Quadrupedal Robots

by

Prajish Kumar Sekoor Lakshmana Sankar

to obtain the degree of

MASTER OF SCIENCE

at the Delft University of Technology,

to be defended publicly on Tuesday October 29, 2019 at 14:00 hours.

Student number:	4743873	
Project duration:	December 1, 2018 – August 31, 2019	
Supervisors:	Marko Bjelonic, Prof. Dr. Marco Hutter, Prof. Dr.-Ing. Heike Vallery,	ETH Zürich ETH Zürich TU Delft
Thesis committee:	Prof. Dr.-Ing. Heike Vallery, Prof. Dr. Ir. Martijn Wisse, Prof. Dr. Ir. A. J. J. (Ton) van den Boom, Prof. Dr. Ir. Arend L. Schwab, Marko Bjelonic,	TU Delft, supervisor TU Delft TU Delft TU Delft ETH Zürich (external)

This thesis is confidential and cannot be made public until October 31, 2019.

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

My year-long association with the Robotics Systems Lab (RSL) of ETH Zürich was remarkably insightful and valuable than my initial expectations. I first thank Dr. Marco Hutter for the acceptance and for arranging an internship in the last minute. Dr. Hutter also organized several events like the Robotics Symposium, RSL Open Lab Day and the RSL summer party, which were great opportunities to network and get a wholesome idea about the research in the lab and outside.

Marko Bjelonic was an amazing mentor in this research endeavour. Marko, who always respects my learning curve, gave me sufficient time to navigate through the monumental pile of code and waited patiently for my contributions. His critical feedback, be it on writing, coding or in general, has always been thorough; that is a great encouragement in itself. Marko and I have spent considerable time putting some sense into the robot: dealing with punctures, smokes and some dangerous self-collisions. With every “why does it do that?” while tuning, I learnt so much with his pragmatic insights and immediate connection to the theory. It is great to work with someone who encourages to persevere hard into one problem, while occasionally asking me to “let go and just chill by the lake”. Thank you, Marko, for that contagious excitement when the robot works and for recommending me to ANYbotics.

I am truly grateful for the involvement of Dr. Heike Vallery in this project. Her reasoning behind ANYmal’s restricted turning abilities while driving, during her visit to RSL, was more concrete than my understanding then. With every conversation and draft review, I am surprised how much my scientific reasoning and communication improves. Thank you, Heike, for your collaboration in the paper and the motivating recommendation letter.

Some encounters in RSL, albeit brief, have helped me at the right times. Thank you, Dario, for helping me get my gears going when I began programming. Yvain, for the useful insights into what works and does not on the robot. Köen, for guiding me on Delft related queries. Fabian, for helping me point out a mistake in my optimization. Thank you Christian and Péter, for introducing me to the locomotion and control team of ANYbotics, where I met EV, Paco and Valentin.

My time at Zürich was way too fun, primarily because of Shruti and Vasily, who were my window to social life. From providing me with good food, fixing my laptop, spontaneous lake dips, hiking, partying and biking, they have pampered me enough. Thank you, Surya, for being my go-to person for rants, advice and laughter; could not have asked for a better roommate and a best friend. Thank you, Jay; we’ve shared numerous moments of success and frustrations throughout this period. Thank you Kishan, Martina, Marti, John, Tuvshee, Anselmo and several other inmates of Tannenrauch 35. I have also been fortunate to receive wonderful encouragements in this period from my friends in Delft: Rick, Anees, Sanjit, Aneesh, Vivek, Gokul, Sarvesh, Krishna, Ashwin, Chris, Alex, Adriana, Niels, Jens and Cathy.

I thank Gerdien for recommending me the *Justus & Louise van Effen Excellence Scholarship* in the first place and also helping in its extension.

My heartfelt gratitude goes to mom and dad for being too supportive and putting my needs beyond theirs several times over these two years. Every time I send them a working video of the robot, they have always shared the same excitement as me in a follow-up call, despite asking “is this really what it is supposed to do?”. *Um, no papa; not really.*

Finally, thank you coffee. **You sustain me.**

Prajish Kumar
4743873
Delft University of Technology
29th October 2019

Table of Contents

I	Introduction	1
II	Motion Planning Overview	2
III	Foot Trajectory Optimization	2
III-A	Preliminary definitions	2
III-B	Coordinate Systems and Reference Frames	2
III-C	Parameterization of Feet Trajectories	3
III-C1	Swing Trajectories	3
III-C2	Stance Trajectories	3
III-D	Formulation of Trajectory Optimization	4
III-E	Objectives	4
III-E1	Minimizing Accelerations	4
III-E2	Avoid Extension of Legs During Stance	5
III-E3	Consistent Trajectories	5
III-E4	Position and Velocity Soft Constraints	5
III-E5	Foothold Predictions	5
III-E6	General	5
III-F	Equality Constraints	6
III-F1	Initial States	6
III-F2	Junction Constraints	6
III-F3	Foothold Constraints	6
III-F4	End Constraints	6
III-G	Inequality Constraints	6
III-H	Switching between Hybrid Walking and Driving	6
IV	Base Trajectory Optimization	6
IV-A	Parameterization Of Base Trajectory	6
IV-B	Objectives	7
IV-C	Equality Constraints	7
IV-D	Inequality Constraints	7
V	Experimental Setup and Implementation	7
V-A	Setup	7
V-B	Implementation	8
V-C	Analysis	8
VI	Results and Discussions	8
VI-A	Gaits	8
VI-B	Solver Times	8
VI-C	Speed and Cost of Transportation	9
VI-D	Reactive Behavior	9
VI-E	Real-World Application	10
VII	Conclusions and Future Work	11
	Appendix A: ANYmal’s ability to make turns without steering the wheels	12
	Appendix B: Representing the objective to avoid extension of legs in stance in QP form	12
	Appendix C: Objectives for minimizing deviation from previous solution	12
	Appendix D: Representing the inequality constraints term in QP form	13
	Appendix E: Shifting from Driving to Walking	13
	References	14

List of Figures

1	Overview of the motion planning and control framework used in our work.	2
2	Top view of the robot showing the coordinate systems and their corresponding reference frames used for the foot optimization.	3
3	All possible sequences of splines for an optimization horizon of gait duration.	4
4	Switching between hybrid walking and pure driving.	6
5	Simulation of ANYmal on wheels while hybrid-crawling in a curve	8
6	ANYmal on wheels reacting to external force disturbances from a human.	10
7	ANYmal on wheels traversing over a wooden plank, on rough terrains and muddy underground environments at the DARPA Subterranean Challenge.	10
8	Measured base and feet trajectories of ANYmal at the DARPA Subterranean Challenge. . . .	11
9	Desired CoM and feet trajectories of ANYmal at the DARPA Subterranean Challenge while hybrid trotting.	11
10	Initial configuration of ANYmal while turning.	12
11	Final configuration of ANYmal while turning.	12

List of Tables

I	Spline durations for all possible spline sequences.	4
II	Planning horizon and the average optimization times for different gaits.	9
III	Maximum heading speeds of the base, v_{\max} , for different gaits	9
IV	Comparison of mechanical COT between multiple gaits	9

Notations

Abbreviations

CoM	Center of Mass
COT	Cost Of Transport
DARPA	Defense Advanced Research Projects Agency
DoF	Degree(s) of Freedom
HAA	Hip Abduction-Adduction
HFE	Hip Flexion-Extension
HO	Hierarchical Optimization
IMU	Inertial Measurement Unit
JPL	Jet Propulsion Laboratory
KFE	Knee Flexion-Extension
LF	Left Front
LH	Left Hind
MPC	Model Predictive Control
NLP	NonLinear Programming
QP	Quadratic Programming
RBDL	Rigid Body Dynamics Library
RF	Right Front
RH	Right Hind
RSL	Robotic Systems Lab
SQP	Sequential Quadratic Programming
TO	Trajectory Optimization
WBC	Whole-Body Controller
w.r.t.	with respect to
ZMP	Zero Moment Point

Math

r	scalar (e.g., radius)
\mathbf{r}	vector (e.g. position vector)
\mathbf{R}	matrix (e.g. rotation matrix)
\mathbf{r}^T	vector transpose
\mathbf{R}^T	matrix transpose
\mathbf{R}^{-1}	matrix inverse
$\dot{\mathbf{r}}$	first derivative of \mathbf{r} w.r.t. t , i.e., $\frac{d\mathbf{r}}{dt}$
$\ddot{\mathbf{r}}$	second derivative of \mathbf{r} w.r.t. t , i.e., $\frac{d^2\mathbf{r}}{dt^2}$
\mathbb{R}	set of all real numbers
$\mathbf{0}$	zero vector/matrix
\mathbb{I}	identity matrix
\parallel	is parallel to

Kinematics and Dynamics

\mathcal{A}	frame of reference
x, y, z	orthogonal unit vectors describing a frame
A	a point in Cartesian space OR a Cartesian coordinate system with origin at A
${}^{\mathcal{A}}r_{BC}$	position vector from point B to point C , represented in \mathcal{A} -frame
${}^{\mathcal{A}}v_B$	velocity vector of point B , represented in \mathcal{A} -frame
${}^{\mathcal{A}}a_B$	acceleration vector of point B , represented in \mathcal{A} -frame
$\mathbf{R}_{\mathcal{X}\mathcal{Y}}$	rotation matrix that transforms a vector in \mathcal{Y} to \mathcal{X} -frame

General

r	position
v	linear velocity
a	linear acceleration
ψ	yaw angle
ω	angular velocity
g, g	gravity vector, magnitude of gravitational acceleration
l	angular momentum
m	mass
n	normal vector to the terrain
t	time
k	time instant

Optimization

ξ, ζ	vector of optimization variables
Q	hessian of the quadratic cost function (in $\frac{1}{2}\zeta^T Q \zeta + c^T \zeta$)
c	linear term of the quadratic cost function (in $\frac{1}{2}\zeta^T Q \zeta + c^T \zeta$)
A	jacobian in the linear equality constraints (in $A \zeta = b$)
b	target values in the linear equality constraints (in $A \zeta = b$)
D	jacobian in the linear inequality constraints (in $D \zeta < e$)
e	maximum values in the linear inequality constraints (in $D \zeta < e$)
w	scalar weight
W	diagonal "weight matrix", with each diagonal element corresponds to the x, y, z components of the vector we seek to minimize

Relevant Publication

Marko Bjelonic, Prajish K. Sankar, Carmine D. Bellicoso, Heike Vallery and Marco Hutter. *Rolling in the Deep - Hybrid Locomotion for Wheeled-Legged Robots using Online Trajectory Optimization*. Submitted to IEEE Robotics and Automation Letters, 2020.

Please cite as

```
@article{bjelonic2020rolling,  
  author = {Bjelonic, Marko and  
            Sankar, Prajish K. and  
            Bellicoso, Carmine D and  
            Vallery, Heike and  
            Hutter, Marco},  
  title  = {Rolling in the Deep - Hybrid Locomotion for Wheeled-Legged  
            Robots using Online Trajectory Optimization},  
  journal = {submitted to IEEE Robotics and Automation Letters},  
  year   = {2020}  
}
```

Paper: <https://arxiv.org/abs/1909.07193>

Video: <https://www.youtube.com/watch?v=ukY0vyM-yfY>

Abstract

Wheeled-legged (hybrid) robots have the potential for highly agile and versatile locomotion in any real-world application requiring rapid, long-distance mobility skills on challenging terrain. The ability to walk and drive simultaneously is an attractive feature of these hybrid systems, but is unexplored in literature.

This report presents an online trajectory optimization framework for high-dimensional wheeled-legged quadrupedal robots where the feet and base trajectories are generated in a model predictive control fashion for robustness against disturbances. Our feet optimization employs a unique parameterization that captures the velocity constraints of the wheels' rolling and our base optimization uses a ZMP-based balance criterion.

Our approach is verified on a torque-controlled quadrupedal robot with nonsteerable wheels. The robot performs hybrid locomotion with different gait sequences on flat and rough terrain. Moreover, our optimization framework generates base trajectories at a rate of about 100Hz and feet trajectories at 1000Hz or higher. In addition, we validated the robotic platform at the Defense Advanced Research Projects Agency (DARPA) Subterranean Challenge, where the robot rapidly maps, navigates, and explores dynamic underground environments.

Trajectory Optimization for Hybrid Walking-Driving Motions on Wheeled Quadrupedal Robots

Prajish Kumar Sankar¹, Marko Bjelonic², Heike Vallery¹ and Marco Hutter²

Abstract—Wheeled-legged (hybrid) robots have the potential for highly agile and versatile locomotion in any real-world application requiring rapid, long-distance mobility skills on challenging terrain. The ability to walk and drive simultaneously is an attractive feature of these hybrid systems, but is unexplored in literature.

This report presents an online trajectory optimization framework for high-dimensional wheeled-legged quadrupedal robots where the feet and base trajectories are generated in a model predictive control fashion for robustness against disturbances. Our feet optimization employs a unique parameterization that captures the velocity constraints of the wheels' rolling and our base optimization uses a ZMP-based balance criterion.

Our approach is verified on a torque-controlled quadrupedal robot with nonsteerable wheels. The robot performs hybrid locomotion with different gait sequences on flat and rough terrain. Moreover, our optimization framework generates base trajectories at a rate of about 100 Hz and feet trajectories at 1000 Hz or higher. In addition, we validated the robotic platform at the Defense Advanced Research Projects Agency (DARPA) Subterranean Challenge, where the robot rapidly maps, navigates, and explores dynamic underground environments.

I. INTRODUCTION

LEGGED robotic systems, on one end, are dexterous enough to navigate in tight spaces and accept terrain discontinuities like gaps, stairs and minor obstacles [1], [2]. On the other end, wheeled mobile robots are an excellent choice for fast locomotion, whose control strategies are fairly established [3]. The recent interest in combining the benefits of both has resulted in hybrid robots¹, which can drive fast on flat terrains and use legged locomotion on rough surfaces.

Compared with traditional legged robots, some hybrid systems have broken speed records and reduced cost of transportation [4]–[6]. Compared with traditional wheeled vehicles, hybrid systems in literature have demonstrated three main advantages. First, the legs can be used as mechanisms to vary footprints for navigating tight spaces without losing balance. Second, the legs can provide suspension for traversing rugged terrains. Most hybrid systems in literature demonstrate just these two benefits [7]–[19].

¹P.K. Sankar and H. Vallery are with Mechanical, Maritime and Materials Engineering (3mE), TU Delft, 2628 CD Delft, Netherlands (email: P.K.Sankar@student.tudelft.nl, h.vallery@tudelft.nl).

²M. Bjelonic and M. Hutter are with the Robotic Systems Lab, ETH Zürich, 8092 Zürich, Switzerland (email: marko.bjelonic@mavt.ethz.ch, mahutter@ethz.ch).

¹Here, we use "hybrid systems" to refer to robots with wheels connected to the main body through extendable/retractable legs.

A few studies have demonstrated the third advantage: the ability to step over obstacles larger than the wheels' radius. A wheeled hexapod in [20] was one of the earliest works to demonstrate walking over large obstacles, but could not be used on-the-fly due to relatively inefficient computations. Walking in [21] was implemented by turning the plane of wheels so that they lay flat on the surface terrain; which is unnecessary considering that the rotation of the wheels could be merely controlled. Some recent works on quadrupedal robots like *CENTAURO* [22], [23], *Momaro* [24], [25] and *ANYmal* [4], [26] have shown impressive results. All these systems demonstrate both driving and walking, but not both simultaneously; the locomotion modes are manually switched depending on the need.

The full exploitation of hybrid systems would be **hybrid motions** where the system drives and walks *at the same time*. Examples include ice skating and roller skating [5]. These hybrid motions, however, are relatively unexplored in literature. *Skaterbots* [27] is an example, but the nonlinear programming (NLP) involved can suffer from the curse of dimensionality when employed on a quadrupedal system with higher degrees of freedom (DoF). Boston Dynamic's Handle [28] is an inspiration but lacks supporting publication.

Motion planning for quadrupedal robots requires generating trajectories for both the feet and the base (torso). These trajectories can be either predesigned, generated based on simple heuristics or using optimal control problems. For *ANYmal* [29] on skates [5], these trajectories are hand-crafted, and hence the robot is generally not robust. In *ANYmal*'s first attempt with actuated wheels [4], the trajectories for the driving motion are generated based on simple heuristics, like velocity projections. This approach works for driving, but cannot be easily extended to hybrid motions.

Trajectory generation by solving an optimal control problem is pervasive and recommended in the literature. Tools like trajectory optimization (TO) and model predictive control (MPC) have produced reliable and robust motions, even for blind locomotion on rough terrains [30]. Currently, only [26] tackles planning through TO for hybrid motions in *ANYmal* on wheels, while incorporating the rolling constraints in the framework. The approach involves generating trajectories for both feet (wheels) and the base in the same optimization problem. This, however, resulted in a slower computation than in [4]. Additionally, the results were verified only on a simulation, and therefore, the extension of their framework to the real robot is unknown.

ANYmal [29] is significantly different from other wheeled quadrupedal robots in literature because the wheels are non-

steerable. The orientation of wheels with respect to (w.r.t.) base follows a sequence of rotations corresponding to hip ad/abduction, hip flex/extension and knee flex/extension²; no endo/exo rotation joint exists for steering. Therefore, ANYmal can only perform small turns while driving. For larger turn angles, the robot must lift its legs and place them somewhere closer, so that the legs do not get overextended. The reasoning behind small turns is in Appendix A.

In this report, we provide a TO framework for hybrid motions on ANYmal on wheels. The framework involves sequential generation of feet and base trajectories in separate optimization problems. We verify the framework on the robot by employing multiple walking gaits with high-speed driving. To the best of our knowledge, this is the first proven attempt of hybrid motions with nonsteerable wheels employed over such a wide range of walking gaits.

II. MOTION PLANNING OVERVIEW

An overview of our planning and control framework is outlined in Fig. 1. The trajectories for the feet and the base are obtained through two separate optimization problems in a sequential way.

First, the foot trajectories are obtained through a quadratic programming (QP) optimization, using external velocity inputs $\mathbf{v}_{\text{des}}, \boldsymbol{\omega}_{\text{des}} \in \mathbb{R}^3$. These inputs can be provided by a user through a joystick or by a navigation planner. Using the predicted foot trajectories, a sequence of support polygons are generated based on the gait. The gait pattern is chosen from a gait library, and contains the timings and the sequence of liftoff and touchdown events. Using this sequence of support polygons, the base trajectory is generated through a NLP optimization.

This sequential generation of feet and base trajectories is expected to be faster than simultaneous optimization [30]. Both the motion plans are produced in a receding horizon fashion, similar to MPC, and an optimization begins as soon as the previous one ends successfully. If our motion planner takes considerably lesser time to generate trajectories than the planning horizon, we believe that the robot can reliably adapt to disturbances, modeling errors and changes in high-level commands, on the fly.

The resulting base and feet trajectories are sent to a whole-body controller (WBC) that converts the variables in the operational/Cartesian space (trajectories) to the joint space (actuator torques).

III. FOOT TRAJECTORY OPTIMIZATION

A. Preliminary definitions

In this report, we define "foot" as the lowermost point of a wheel, below the center of its axis, which would be the point of contact with the ground if the leg was in stance. In our implementation, this point is assumed fixed to the leg's shank and independent of the wheel's rotation.

²This sequence describes the orientation of the leg's shank. Since the wheels cannot steer w.r.t. to the shank, this sequence describes the orientation of the wheel's plane as well.

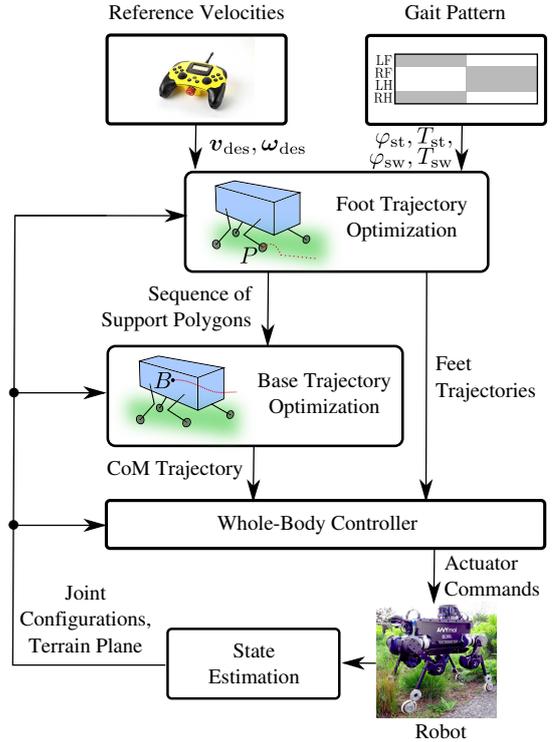


Fig. 1. Overview of the motion planning and control framework used in our work.

Since we simplified the rigid body of a wheel to a point, we will be optimizing for the position coordinates of the foot, as explained in the following subsections.

B. Coordinate Systems and Reference Frames

We distinguish between reference frames and coordinate systems in such a way that reference frames encode the axes directions (as unit vectors), while coordinate systems encode axes directions as well as an origin. Therefore, multiple coordinate systems can be associated with the same reference frame.

All coordinate systems and frames of reference used for planning the foot trajectories, shown in Fig. 2, are represented w.r.t. a fixed inertial "world" $W\{x_W, y_W, z_W\}$ -frame. The pose of the robot's torso is described using a body-fixed coordinate system at its center of mass (CoM), B . This coordinate system is associated with a "base" B -frame with x_B along its heading direction. The high-level velocity commands $(\mathbf{v}_{\text{des}}, \boldsymbol{\omega}_{\text{des}})$ are provided in a "control" C -frame, with x_C along the projection of base heading direction (x_B) on the terrain and z_C along the terrain normal. The origin of the corresponding coordinate system, C , coincides with that of the world frame.

The trajectory of a foot is generated and expressed in a coordinate system P , associated with a "plan" P -frame, oriented parallel to the control frame. The origin of P is at the projection of the wheel's axis center on the terrain. In ANYmal, each wheel is pre-aligned such that its rotation axis is along y_P .

Let B' and P' be the future positions, and frames B' and P' show the future orientations of the base and the foot

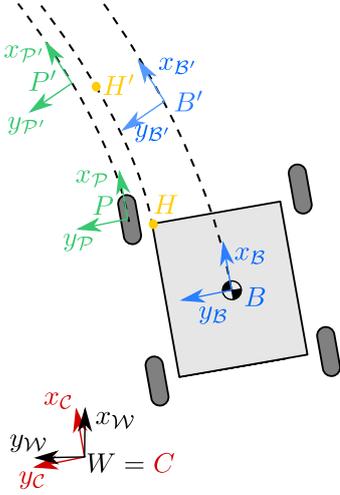


Fig. 2. Top view of the robot showing the coordinate systems and their corresponding reference frames used for the foot optimization.

respectively, after some time has passed. In this time interval, we assume the ‘hip’, H , of the robot has moved to H' .

If the pitch and roll angles of the base w.r.t. the ground plane are set to zero, then the base frame, control frame and plan frame become identical, as explained in [26]. So, as long as this condition is met, the wheel heading direction is always parallel to the base heading direction ($x_{P'} \parallel x_{B'}$). Thus, when the base is at B' , the foot, now at P' , will have its wheel rotation axis (and hence the direction of no side slip) along $y_{P'}$.

Additionally, since the axes z_P and $z_{P'}$ are parallel at all times, the orientation of P' w.r.t P ($\mathbf{R}_{P'P}$) at time t is a function of yaw angle ($\psi_{\text{des}} = \omega_{\text{des}}^z t$) alone, expressed as,

$$\mathbf{R}_{P'P}(t) = \begin{bmatrix} \cos(\omega_{\text{des}}^z t) & -\sin(\omega_{\text{des}}^z t) & 0 \\ \sin(\omega_{\text{des}}^z t) & \cos(\omega_{\text{des}}^z t) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

Here, $\omega_{\text{des}}^z \in \mathbb{R}$ is the desired base yaw rate given as a high-level command in \mathcal{C} -frame. For ease of calculations, it is assumed constant over the optimization horizon. This is a valid assumption if it changes in a much slower rate than the update frequency of our optimization.

C. Parameterization of Feet Trajectories

The feet trajectories are described as a sequence of continuous and smooth splines. A stance segment is allocated one spline, whereas a swing segment is allocated two, which connect at the point where the foot reaches the maximum height from the ground.

1) Swing Trajectories: The x, y and z components of a swing spline are parameterized as quintic polynomials in time t , as in [31]–[33]. A quintic polynomial is chosen over polynomials of lower degree because they provide more freedom to change the shape of the trajectory mid-swing, which is beneficial for recovery during disturbances. It is the same reason why we choose two splines for a swing instead of just one.

The x component of the foot position in P coordinate frame during swing can be written as $\eta(t)\alpha_x$, where,

$$\eta(t) = \begin{bmatrix} t^5 & t^4 & t^3 & t^2 & t & 1 \end{bmatrix}, \quad (2)$$

$$\alpha_x = \begin{bmatrix} \alpha_5^x & \alpha_4^x & \alpha_3^x & \alpha_2^x & \alpha_1^x & \alpha_0^x \end{bmatrix}^T.$$

Using this, the position of the foot as a function of time, represented in P coordinate frame, ${}^P r_{PP'}(t) \in \mathbb{R}^3$, can be written as,

$${}^P r_{PP'}(t) = \begin{bmatrix} \eta(t) & \mathbf{0}_{6 \times 1} & \mathbf{0}_{6 \times 1} \\ \mathbf{0}_{6 \times 1} & \eta(t) & \mathbf{0}_{6 \times 1} \\ \mathbf{0}_{6 \times 1} & \mathbf{0}_{6 \times 1} & \eta(t) \end{bmatrix} \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} = \mathbf{T}_{\text{sw}}(t)\zeta_{\text{sw}}, \quad (3)$$

with $\mathbf{T}_{\text{sw}}(t) \in \mathbb{R}^{3 \times 18}$, the time matrix of a swing spline. The vector $\zeta_{\text{sw}} = [\alpha_x \ \alpha_y \ \alpha_z]^T \in \mathbb{R}^{18}$ contains the polynomial coefficients describing all three components of the swing spline. Velocity and acceleration of the foot in P -frame are given by ${}^P v_{P'}(t) = \dot{\mathbf{T}}_{\text{sw}}(t)\zeta_{\text{sw}}$ and ${}^P a_{P'}(t) = \ddot{\mathbf{T}}_{\text{sw}}(t)\zeta_{\text{sw}}$, respectively.

2) Stance Trajectories: The parameterization of stance splines can be simplified greatly. Firstly, the z component of the foot trajectory in P coordinate system, during stance, is zero. Secondly, unlike swing splines, a great degree of precision is not expected for stance trajectories. If the robot experiences any disturbance, the legs adapt primarily by swinging. Legs in stance do not contribute to recovery as much. Therefore, parameterizing stance trajectories as cubic polynomials in time is sufficient³.

Thirdly, we incorporate the velocity constraint, corresponding to the ‘no-side-slip’ condition of the wheel, in the stance parameterization, as in [26], instead of imposing them as a hard constraint for every time step in the optimization problem. This way, the constraints from the wheels are taken into account even before the optimization begins. Also, the motion along x and y direction become coupled. This reduces the number of optimization variables and the number of equality constraints, which would help in improving the solver times.

This no-side-slip constraint of the wheel can be used to parameterize the velocities, instead of the position trajectories, as follows. Let the foot at time t be at P' with the wheel’s heading direction along $x_{P'}$. For the no-slip condition, we need the velocity of the foot to have a component along $x_{P'}$ alone. We represent this x component of velocity in P' -frame as a quadratic polynomial in t , as $\beta_0 + \beta_1 t + \beta_2 t^2$. Meanwhile the velocities along $y_{P'}$ and $z_{P'}$ are set to zero. To represent this velocity in P -frame from P' -frame, we use the rotation matrix $\mathbf{R}_{PP'}(t)$ (from (1)) as,

$${}^P v_{P'}(t) = \mathbf{R}_{PP'}(t) \begin{bmatrix} 1 & t & t^2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}. \quad (4)$$

³We avoid polynomials of order lower than three as they restrict the accelerations too much.

To get the foot trajectory, we integrate this velocity w.r.t t and add the initial position $[x_{\text{init}} \ y_{\text{init}} \ 0]^T$ as,

$$\mathcal{P}\mathbf{r}_{PP'}(t) = \begin{bmatrix} x_{\text{init}} \\ y_{\text{init}} \\ 0 \end{bmatrix} + \int_0^t \mathcal{P}\mathbf{v}_{P'}(t)dt. \quad (5)$$

Assuming ω_{des}^z is constant over the optimization horizon, the integration is solved analytically. The resulting expression is,

$$\mathcal{P}\mathbf{r}_{PP'}(t) = \mathbf{T}_{\text{st}}(\omega_{\text{des}}^z, t)\zeta_{\text{st}}, \quad (6)$$

where, $\mathbf{T}_{\text{st}}(\omega_{\text{des}}^z, t) \in \mathbb{R}^{3 \times 5}$ is the time matrix of the stance spline and the vector $\zeta_{\text{st}} = [\beta_0 \ \beta_1 \ \beta_2 \ x_{\text{init}} \ y_{\text{init}}]^T \in \mathbb{R}^5$ contains the polynomial coefficients that describes this stance spline.

Unlike in swing, the position of the foot in stance at time t depends on ω_{des}^z too. For brevity and consistency, we write the foot trajectories in stance as $\mathcal{P}\mathbf{r}_{PP'}(t)$, instead of $\mathcal{P}\mathbf{r}_{PP'}(\omega_{\text{des}}^z, t)$. For negligible base yaw rates, $|\omega_{\text{des}}^z| < 10^{-5}$ rad/s, we use the time matrix of the limiting case, $\mathbf{T}_{\text{st}}(t) = \lim_{\omega_{\text{des}}^z \rightarrow 0} \mathbf{T}_{\text{st}}(\omega_{\text{des}}^z, t)$.

D. Formulation of Trajectory Optimization

The optimization is performed over a constant horizon equal to the duration of the chosen gait, T , which is the sum of stance (T_{st}) and swing (T_{sw}) durations. Depending on the current stance or swing phase, denoted by $\varphi_{\text{st}} \in [0, 1)$ and $\varphi_{\text{sw}} \in [0, 1)$ respectively, the sequence and the durations of the splines can be predicted. Fig. 3 shows the three possible spline sequences and Table I explains how the durations are computed.

The spline coefficients of all swing (ζ_{sw}) and stance (ζ_{st}) segments in a spline sequence are added to a vector $\zeta \in \mathbb{R}^{n_s}$,

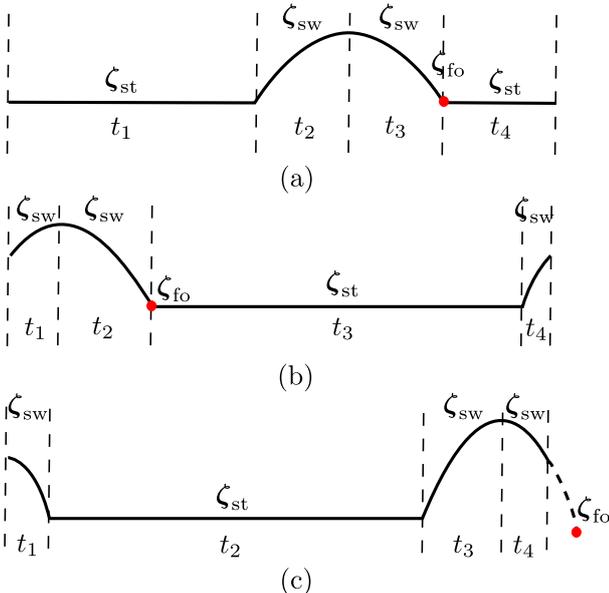


Fig. 3. All possible sequences of splines for an optimization horizon of gait duration. The variables t_1, \dots, t_4 are the durations of the individual splines, which can be zero. The red dot shows the position that ζ_{fo} describes.

TABLE I

THE SPLINE DURATIONS FOR ALL POSSIBLE SPLINE SEQUENCES SHOWN IN FIG. 3.

Spline Sequence	(a)	(b)	(c)
t_1	$(1 - \varphi_{\text{st}})T_{\text{st}}$	$(0.5 - \varphi_{\text{sw}})T_{\text{sw}}$	$(1 - \varphi_{\text{sw}})T_{\text{sw}}$
t_2	$T_{\text{sw}}/2$	$T_{\text{sw}}/2$	T_{st}
t_3	$T_{\text{sw}}/2$	T_{st}	$T_{\text{sw}}/2$
t_4	$\varphi_{\text{st}}T_{\text{st}}$	$\varphi_{\text{sw}}T_{\text{sw}}$	$(\varphi_{\text{sw}} - 0.5)T_{\text{sw}}$

which contains all the variables to optimize for. This vector also includes $\zeta_{\text{fo}} \in \mathbb{R}^2$ that describes the x, y positions of the predicted footholds for swing touchdown. Depending on the spline sequence, the number of optimization variables, n_s , vary.

The problem of finding the trajectory of a foot is formulated as a QP problem of the form,

$$\begin{aligned} \min_{\zeta} \quad & \frac{1}{2}\zeta^T \mathbf{Q}\zeta + \mathbf{c}^T \zeta \\ \text{s.t.} \quad & \mathbf{A}\zeta = \mathbf{b}, \\ & \mathbf{D}\zeta < \mathbf{e}. \end{aligned} \quad (7)$$

The terms that contribute to the cost function hessian, $\mathbf{Q} \in \mathbb{R}^{n_s \times n_s}$, and the linear term $\mathbf{c} \in \mathbb{R}^{n_s}$ are explained in Sec. III-E. In Sec. III-F, the terms contributing to the equality constraint Jacobian $\mathbf{A} \in \mathbb{R}^{n_e \times n_s}$ and the corresponding vector of target values $\mathbf{b} \in \mathbb{R}^{n_e}$ are explained, where n_e is the total number of equality constraints. Finally, Sec. III-G explains how the inequality constraint jacobian $\mathbf{D} \in \mathbb{R}^{n_i \times n_s}$ and the corresponding vector of maximum values $\mathbf{e} \in \mathbb{R}^{n_i}$ are obtained, where n_i is the total number of inequality constraints.

E. Objectives

The quadratic cost function in (7) is obtained as a weighted sum of various objectives. The following subsections describe each objective as a task and their purpose in generating a realistic foot trajectory. These individual objectives are quadratic in ζ , therefore, \mathbf{Q} in (7) is the sum of all the individual cost function Hessians and \mathbf{c} is the sum of the individual linear terms as well.

Some of the following objective terms use scalar weights, $w \in \mathbb{R}$, and others use weight matrices, $\mathbf{W} \in \mathbb{R}^{3 \times 3}$. A weight matrix is a diagonal matrix with its diagonal elements containing weights for each component of a vector we seek to minimize.

1) *Minimizing Accelerations*: Minimizing the total acceleration of the feet, as in [4], [34], can help reduce jerky motions. For a swing spline of duration t_f seconds, the cost function term takes the form,

$$\frac{1}{2}\zeta_{\text{sw}}^T \left(\underbrace{2 \int_0^{t_f} \ddot{\mathbf{T}}_{\text{sw}}^T(\tau) \mathbf{W}_{\text{acc,sw}} \ddot{\mathbf{T}}_{\text{sw}}(\tau) d\tau}_{\mathbf{Q}_{\text{acc,sw}}} \right) \zeta_{\text{sw}}, \quad (8)$$

where $\mathbf{W}_{\text{acc,sw}}$ is a weight matrix and $\mathbf{c}_{\text{acc,sw}} = \mathbf{0}_{18 \times 1}$.

For a stance segment with duration t_f , we follow a similar formulation with a weight matrix $\mathbf{W}_{\text{acc,st}}$ as,

$$\frac{1}{2} \zeta_{\text{st}}^T \underbrace{\left(2 \int_0^{t_f} \ddot{\mathbf{T}}_{\text{st}}^T(\omega_{\text{des}}^z, \tau) \mathbf{W}_{\text{acc,st}} \ddot{\mathbf{T}}_{\text{st}}(\omega_{\text{des}}^z, \tau) d\tau \right)}_{\mathbf{Q}_{\text{acc,st}}} \zeta_{\text{st}}, \quad (9)$$

with $\mathbf{c}_{\text{acc,st}} = \mathbf{0}_{18 \times 1}$ as well.

2) *Avoid Extension of Legs During Stance*: During stance, differences in the heading velocities of a wheel and the base can lead to configurations where the corresponding foot is either too forward or lagging behind w.r.t. the base. To avoid this unnecessary extension of a leg, we penalize the deviation of the foot's position from a reference foothold w.r.t. base, in the heading ($x_{\mathcal{P}'}$) direction. This reference is the projected hip position on the ground, obtained from the robot's default standing configuration.

If D is the reference position of the foot w.r.t. base and D' , the future position of the same, we seek to minimize the x component of the distance between D' and predicted foot position P' in \mathcal{P}' -frame at all time steps, $k = 1, 2, \dots, N$ with a weight w_{def} , i.e.,

$$\sum_{k=1}^N w_{\text{def}} \left\| \mathcal{P}' \mathbf{r}_{\mathcal{P}'D'}^x(t_k) \right\|^2 \Delta t, \quad \forall t \in [0, t_f], \quad (10)$$

where t_f is the duration of the stance spline, t_k the time at time step k and $\Delta t = t_k - t_{k-1}$. The formulation of this term as a QP objective function is detailed in Appendix B.

3) *Consistent Trajectories*: For fast motion planners that run several times a second, large deviations between successive solutions can produce quivering motions. To avoid this, we add a cost function term where the deviations between the kinematic states from the current solution ζ_i and from that of the previous solution ζ_{i-1} are minimized. If t' is the time elapsed since last optimization, the cost function term for minimizing the position deviation over the entire optimization horizon, looks like

$$\sum_{k=1}^N \left\| \mathcal{P} \mathbf{r}_{\mathcal{P}P'}^i(t_k) - \mathcal{P} \mathbf{r}_{\mathcal{P}P'}^{i-1}(t_k + t') \right\|_{\mathbf{W}_{\text{pre}}^{\text{pos}}}^2 \Delta t, \quad \forall t_k \in [0, T], \quad (11)$$

with a weight matrix, $\mathbf{W}_{\text{pre}}^{\text{pos}}$. Here, $\mathcal{P} \mathbf{r}_{\mathcal{P}P'}^{i-1}$ is the position obtained from the previous solution, ζ_{i-1} . Objectives for minimizing velocity and acceleration deviations are added in a similar formulation, as explained in Appendix C.

4) *Position and Velocity Soft Constraints*: Positions and velocities at certain points of the trajectory can be set to desired values using soft constraints when it is acceptable to deviate if necessary.

For trajectories that begin with a stance segment, we set the x component of the initial velocity to the reference heading velocity of the torso, $v_{\text{des}}^x \in \mathbb{R}$, which is obtained from an external source like a joystick or a navigation planner. We believe that giving the initial velocity of stance as a soft constraint rather than a hard constraint can avoid wheel slip

and skid. With a weight of w_{vel} , we minimize the norm $\| \mathcal{P} \mathbf{v}_{\mathcal{P}'}^x(0) - v_{\text{des}}^x \|^2$. This expands to,

$$\frac{1}{2} \zeta_{\text{st}}^T \underbrace{(2w_{\text{vel}} \mathbf{\Gamma}^T \mathbf{\Gamma})}_{\mathbf{Q}_{\text{vel}}} \zeta_{\text{st}} + \underbrace{(-2w_{\text{vel}} v_{\text{des}}^x \mathbf{\Gamma})}_{\mathbf{c}_{\text{vel}}^T} \zeta_{\text{st}}, \quad (12)$$

where $\mathbf{\Gamma} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \dot{\mathbf{T}}_{\text{st}}(\omega_{\text{des}}^z, 0)$.

Likewise, we set the maximum height of the swing to a pre-defined value, h_{max} . For instance, consider the spline sequence in Fig. 3(b), with the first swing spline having a duration of t_1 . The cost function term with a weight of w_{sh} can be written as $w_{\text{sh}} \left\| \mathcal{P} \mathbf{r}_{\mathcal{P}'}^z(t_1) - h_{\text{max}} \right\|^2$, which can be expanded as,

$$\frac{1}{2} \zeta_{\text{sw}}^T \underbrace{(2w_{\text{sh}} \mathbf{\Gamma}^T \mathbf{\Gamma})}_{\mathbf{Q}_{\text{sh}}} \zeta_{\text{sw}} + \underbrace{(-2w_{\text{sh}} h_{\text{max}} \mathbf{\Gamma})}_{\mathbf{c}_{\text{sh}}^T} \zeta_{\text{sw}}, \quad (13)$$

with $\mathbf{\Gamma} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{T}_{\text{sw}}(t_1)$.

5) *Foothold Predictions*: The variables in ζ_{fo} , representing the x, y positions at swing touchdown, are set to a predicted foothold position $\mathbf{r}_{\text{foot}} \in \mathbb{R}^3$, as,

$$\left\| \zeta_{\text{fo}} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{r}_{\text{foot}} \right\|_{\mathbf{W}_{\text{foot}}}^2 \quad (14)$$

with a weight matrix of $\mathbf{W}_{\text{foot}} \in \mathbb{R}^{2 \times 2}$. This is formulated as a soft constraint, as opposed to a hard constraint, to avoid conflicts with kinematic bounds for the foot, which is formulated as an inequality constraint in Sec. III-G.

The position \mathbf{r}_{foot} , in coordinate frame P , is calculated as a sum of a "feedforward" velocity projection term $\mathbf{r}_{\text{vel}} \in \mathbb{R}^3$ and a "feedback" term for handling disturbances mid-swing $\mathbf{r}_{\text{inv}} \in \mathbb{R}^3$, $\mathbf{r}_{\text{foot}} = \mathbf{r}_{\text{vel}} + \mathbf{r}_{\text{inv}}$. The feedforward term is predicted based on the given high-level velocity commands ($\mathbf{v}_{\text{des}}, \boldsymbol{\omega}_{\text{des}} \in \mathbb{R}^3$) and the time remaining in swing, Δt_s as

$$\mathbf{r}_{\text{vel}} = \mathbf{r}_{\text{def}} + (\mathbf{v}_{\text{des}} + \boldsymbol{\omega}_{\text{des}} \times \mathbf{r}_{BP}) \Delta t_s, \quad (15)$$

where, $\mathbf{r}_{\text{def}} \in \mathbb{R}^3$ is the default foot position from standing configuration and $\mathbf{r}_{BP} \in \mathbb{R}^3$ the projection of the position vector from base to the current foot position on the terrain.

The feedback term, \mathbf{r}_{inv} , is adapted from the concept of capture point [35] that uses a simplified linear inverted pendulum model to determine where the foot should be placed for the robot to come to a complete halt. Extending this concept to legged robots as in [31], based on Raibert's flight controller [36], this feedback term is calculated as

$$\mathbf{r}_{\text{inv}} = k_{\text{inv}} (\mathbf{v}_{\text{ref}}^d - \mathbf{v}_{\text{ref}}) \sqrt{\frac{h}{g}}, \quad (16)$$

where $\mathbf{v}_{\text{ref}}^d \in \mathbb{R}^3$ is the desired and $\mathbf{v}_{\text{ref}} \in \mathbb{R}^3$ the measured reference velocity between associated hip and middle of the torso, h the height of the hip above the ground, g the gravitational acceleration and k_{inv} the gain for balancing.

6) *General*: QP solvers that use the Active-Set algorithm demand that the hessian, \mathbf{Q} , be positive definite [37]. This could be violated if the hessian contains rows or columns with just zeros. Therefore, as a final step, we add a regularizing matrix, $\mathbf{Q}_{\text{reg}} = \rho \mathbb{I}_{n_s \times n_s}$, to the hessian, where \mathbb{I} is an identity matrix and $\rho = 10^{-8}$.

F. Equality Constraints

1) *Initial States:* The initial position of the trajectory is set to $[x_{\text{init}} \ y_{\text{init}} \ 0]^T$ as,

$$\mathbf{T}_{\text{sw}}(0)\zeta_{\text{sw}} = [x_{\text{init}} \ y_{\text{init}} \ 0]^T, \quad (17)$$

where the value of y_{init} is obtained from the measured position of the foot, while x_{init} is obtained from a fused state of the measured position and the position obtained from the previous solution.

Likewise, for trajectories that start with a swing spline, we set the initial velocity and acceleration as hard constraints. The initial velocity is set to a fused state of measured velocity and from that of previous solution. Since we do not obtain acceleration measurements, the initial acceleration is set to that from previous solution alone.

2) *Junction Constraints:* We constrain the position, velocity and acceleration at the junction of two swing splines characterized by their spline coefficients $\zeta_{\text{sw},1}$ and $\zeta_{\text{sw},2}$ as,

$$\begin{bmatrix} -\mathbf{T}_{\text{sw},1}(t_{\text{sw}}) & \mathbf{T}_{\text{sw},2}(0) \\ -\dot{\mathbf{T}}_{\text{sw},1}(t_{\text{sw}}) & \dot{\mathbf{T}}_{\text{sw},2}(0) \\ -\ddot{\mathbf{T}}_{\text{sw},1}(t_{\text{sw}}) & \ddot{\mathbf{T}}_{\text{sw},2}(0) \end{bmatrix} \begin{bmatrix} \zeta_{\text{sw},1} \\ \zeta_{\text{sw},2} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}, \quad (18)$$

where t_{sw} is the duration of the first swing spline.

At stance-swing junctions, we avoid constraining of acceleration as it can hinder liftoff and touchdown. The constraints are formulated as,

$$\begin{bmatrix} -\mathbf{T}_{\text{st}}(\omega_{\text{des}}^z, t_{\text{st}}) & \mathbf{T}_{\text{sw}}(0) \\ -\dot{\mathbf{T}}_{\text{st}}(\omega_{\text{des}}^z, t_{\text{st}}) & \dot{\mathbf{T}}_{\text{sw}}(0) \end{bmatrix} \begin{bmatrix} \zeta_{\text{st}} \\ \zeta_{\text{sw}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}, \quad (19)$$

where t_{st} is the stance spline duration.

3) *Foothold Constraints:* If ζ_{sw} characterizes the spline that represents the second half of a swing, then the end position (at spline duration t_{sw}) is constrained to ζ_{fo} as,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{T}_{\text{sw}}(t_{\text{sw}})\zeta_{\text{sw}} = \zeta_{\text{fo}}. \quad (20)$$

The end of the spline that represents the first half of the swing is set to the point where the foot reaches maximum height. If $\mathbf{r}_{\text{init}} \in \mathbb{R}^2$ is the initial x, y position of this spline (with a duration of $t_{\text{sw},1}$), and $t_{\text{sw},2}$ the duration of the following spline representing the second half of the swing, the final point of the first swing spline is given by,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{T}_{\text{sw},1}(t_{\text{sw},1})\zeta_{\text{sw},1} = \frac{t_{\text{sw},1}}{t_{\text{sw},1} + t_{\text{sw},2}} \zeta_{\text{fo}} + \frac{t_{\text{sw},2}}{t_{\text{sw},1} + t_{\text{sw},2}} \mathbf{r}_{\text{init}}. \quad (21)$$

4) *End Constraints:* When the spline sequence ends with a swing, the z component of the final position of that spline is set to the current height of the foot from the ground, whereas the x, y components are found using velocity projection, as in (15). With $\Delta t_s = t_f$, the duration of the final swing spline, this constraint is formulated as,

$$\mathbf{T}_{\text{sw}}(t_f)\zeta_{\text{sw}} = \begin{bmatrix} \mathbf{r}_{\text{vel}}^x & \mathbf{r}_{\text{vel}}^y & \mathcal{P}\mathbf{r}_{\text{PP}'}^z(0) \end{bmatrix}^T. \quad (22)$$

G. Inequality Constraints

To avoid overextension of legs from the base, we provide kinematic bounds for the feet. This is formulated as inequality constraints, where the foot position P' , at time $t \in [0, T]$, is limited to a rectangle of size $2x_{\text{offset}} \times 2y_{\text{offset}}$ centered around the projection of the corresponding hip position on the terrain, H' . This implies,

$$\begin{aligned} |\mathcal{P}'\mathbf{r}_{P'H'}^x(t_k)| &< x_{\text{offset}} \quad \forall t_k \in [0, T], \\ |\mathcal{P}'\mathbf{r}_{P'H'}^y(t_k)| &< y_{\text{offset}} \quad \forall t_k \in [0, T], \end{aligned} \quad (23)$$

where the formulation of this term as an inequality constraint in (7) is detailed in Appendix D.

H. Switching between Hybrid Walking and Driving

For smaller reference angular velocities, ω_{des}^z , the robot can continue driving. For larger velocities, we run into the problem of leg extension. In an optimization horizon of T seconds, the foot would have moved relative to the base from the current measured position, $(x_{\text{mea}}, y_{\text{mea}})$, as shown in Fig. 4. The robot can continue driving if this predicted foot position after T lies within the circle of radius l_{max} centered around the default foot positions $(x_{\text{def}}, y_{\text{def}})$ from the standing configuration. Therefore, the condition for switching from driving to hybrid walking can be formulated as,

$$\begin{aligned} (x_{\text{mea}} - x_{\text{des}} + y_{\text{mea}}\omega_{\text{des}}^z T)^2 \\ + (y_{\text{mea}} - y_{\text{des}} - x_{\text{mea}}\omega_{\text{des}}^z T)^2 \geq l_{\text{max}}^2. \end{aligned} \quad (24)$$

If the condition is violated even for a foot, we switch to walking. The derivation of this condition is given in Appendix E.

IV. BASE TRAJECTORY OPTIMIZATION

A. Parameterization Of Base Trajectory

To predict the trajectory of the torso's CoM, we use the motion optimization scheme outlined in [4], [34]. The trajectories are generated in a new base-"plan" coordinate frame, $\mathcal{P}\{z_{\mathcal{P}}, y_{\mathcal{P}}, z_{\mathcal{P}}\}$, located at the footprint center and aligned along the estimated terrain. The x, y, z coordinates of the CoM trajectory are represented as a sequence of splines parameterized as quintic polynomials in time. The spline coefficients are stacked in ξ , which contains all the optimization variables.

The continuous feet trajectories from the foot optimization are used to generate a sequence of support polygons at

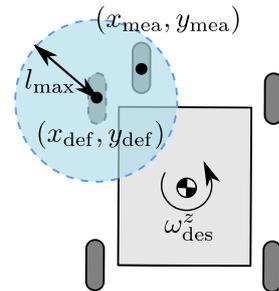


Fig. 4. Switching between hybrid walking and pure driving. Top view of the robot showing the maximum allowable leg extension l_{max} , which triggers the switch between the two modes.

multiple time instants over the optimization horizon. Three splines are allocated for every consecutive pair of support polygons, corresponding to the three DoF of the base we are optimizing for. Therefore, for n_p support polygons over the horizon, we require $3(n_p - 1)$ splines. Since each spline has 6 polynomial coefficients, the size of ξ is $18(n_p - 1)$. If the gait includes full-flight phases, then we reserve two splines for each consecutive pair of support polygons and the size of ξ increases accordingly.

B. Objectives

Some of the objectives that contribute to the cost function for the base optimization are similar to that of foot optimization. Over the optimization horizon, we minimize the acceleration of the base for smooth motions, as in (8) [38]. Moreover, the deviations in positions, velocities and accelerations between successive optimization solutions are minimized, as in (11).

Additionally, we penalize deviations of the base trajectory from a reference trajectory, or a *path regularizer*, $\pi(t)$. This path regularizer is obtained as an approximate trajectory for the base from the external high-level velocities, as in [34]. Finally, as a terminal cost, the end of the trajectory is set as a soft constraint to the center of the final support polygon.

The resulting cost function of the base optimization is quadratic in ξ [34].

C. Equality Constraints

The initial states of the trajectory are set as hard constraints, linear in ξ , similar to our approach for foot optimization. The initial position and velocity are set to fused states obtained from the current measurements and previous solution, while the acceleration is constrained to the state obtained from the previous solution alone.

For continuity, linear constraints are set for kinematic states at the junction between splines as in [34]. While the constraints are set for position and velocity for all junctions, we constrain the acceleration only if the splines belong to two intersecting support polygons.

D. Inequality Constraints

To ensure stability in the sense of balance, we constrain the zero moment point (ZMP) [39] to lie within the support polygon at all time steps in the optimization horizon, T [40]. This can be formulated as inequality constraints of the form,

$$\begin{bmatrix} p & q & 0 \end{bmatrix} \mathbf{r}_{\text{ZMP}}(t_k) + r < 0, \quad \forall t_k \in [0, T], \quad (25)$$

where $\mathbf{d} = [p \ q \ r]^T$ describes the edges of the support polygon with added safety margins at its boundaries.

The position of the ZMP at time t , $\mathbf{r}_{\text{ZMP}}(t) \in \mathbb{R}^3$, is a function of the position of CoM, $\mathbf{r}_{\text{CoM}}(t)$, as,

$$\mathbf{r}_{\text{ZMP}}^{x,y}(t) = \frac{\mathbf{n} \times (m\mathbf{r}_{\text{CoM}}(t) \times (\mathbf{g} - \ddot{\mathbf{r}}_{\text{CoM}}(t)) - \dot{\mathbf{l}}(t))}{\mathbf{n} \cdot (m\mathbf{g} - m\ddot{\mathbf{r}}_{\text{CoM}}(t))}, \quad (26)$$

where $\mathbf{n} \in \mathbb{R}^3$ is the terrain normal, $m \in \mathbb{R}$ the mass of the base, $\mathbf{g} \in \mathbb{R}^3$ the gravity vector and $\mathbf{l} \in \mathbb{R}^3$ the angular

momentum of the base at the CoM. Since the ZMP position is nonlinear in the CoM position $\mathbf{r}_{\text{CoM}}(t)$, it is nonlinear in ξ as well. Substituting (26) in (25), the inequality constraints, likewise, are nonlinear in ξ .

If the robot walks on a horizontal flat terrain ($\mathbf{n} \parallel \mathbf{g}$) with a constant angular momentum ($\dot{\mathbf{l}} = 0$) and height of the base from the ground ($\dot{\mathbf{r}}_{\text{CoM}}^z = 0$), (26) can be simplified greatly to produce an expression linear in the x, y components of \mathbf{r}_{CoM} as,

$$\mathbf{r}_{\text{ZMP}}^{x,y}(t) = \mathbf{r}_{\text{CoM}}^{x,y}(t) - \frac{\mathbf{r}_{\text{CoM}}^z(t)}{g} \ddot{\mathbf{r}}_{\text{CoM}}^{x,y}(t), \quad (27)$$

where g is the magnitude of the gravitational acceleration. This equation can be extended for inclined surfaces by replacing g with the component of gravity along the terrain normal and adding its other components to the accelerations $\ddot{\mathbf{r}}_{\text{CoM}}^{x,y}$ respectively.

Through these assumptions, we have treated the robot as a linear inverted pendulum (or a cart-table) model [41], [42]. Substituting this in (25), the inequalities become linear in ξ .

In our implementation, the convexity of the base optimization problem depends on the linearity of the inequality constraints in ξ . As long as we employ gaits like driving, crawling or trotting on flat surfaces, the assumptions for the cart-table model are valid. This maintains the base optimization convex, guaranteeing global minima and fast computations.

V. EXPERIMENTAL SETUP AND IMPLEMENTATION

A. Setup

We tested the proposed optimization framework on ANYmal [4], a quadrupedal robot equipped with nonsteerable, torque-controlled wheels as end-effectors. The robot receives external velocity inputs from a joystick, controlled by the user. The computations were carried out on a PC (Intel i7-7500U, 2.7 GHz, dual-core 64-bit) integrated into the robot. For applications that require autonomous navigation, the velocity commands were obtained from a navigation planner, where the higher-level computations (such as perception, mapping, localization, path planning, path following, and object detection) were carried out on additional PCs in the robot.

The optimization was implemented in C++ using open-source libraries like *Eigen* [43] for linear algebraic operations and *QuadProg++* [44] for solving QP based on Goldfarb-Ignani Active-Set algorithm [37]. The base TO uses a custom sequential quadratic problem (SQP) algorithm, which solves the NLP problem by iterating through a sequence of QP problems. Rigid Body Dynamics Library, *RDBL* [45] and *Kindr* [46] were used for kinematics and dynamics, while simulating in *Gazebo* [47] with Open Dynamics Engine, *ODE* [48] as a physics engine.

For tracking the operational space reference variables, we use a WBC, inspired from [49], that generates the torque commands for the joints by solving a sequence of prioritized constraints. The WBC is extended for ANYmal on wheels, as described in [4], incorporating the additional rolling constraint associated with the wheels.

The feet TO, base TO and the WBC run parallel in multiple threads. The WBC and the state estimation (as established in

[49], [50] and extended for ANYmal on wheels as in [4] run in a fixed 400 Hz loop. The trajectory optimizations, however, are not bound by time. Therefore, the observed frequencies of these optimizations reflect their solver times. If either of the foot or base optimization fails, we use the solution from the corresponding previously successful optimization.

B. Implementation

As in [26], we set the pitch and roll of the base w.r.t. \mathcal{W} -frame to zero at all times. This is crucial for our initial assumption that the heading directions of the base and wheels are always the same.

The external commands for linear velocity $\mathbf{v}_{\text{des}} = [v_{\text{des}}^x \ v_{\text{des}}^y \ 0]^T$ and the angular velocity $\boldsymbol{\omega}_{\text{des}} = [0 \ 0 \ \omega_{\text{des}}^z]^T$ of the base are given in the "control" \mathcal{C} -frame. Although additional velocity commands, like v_{des}^z and ω_{des}^{xy} , can provide more functionality to our robot, our options for these commands are limited by most off-the-shelf joysticks.

In the foot TO, some objective terms, like (10) and (11), as well as the inequality constraints in (23), require discretizing the spline segments in intervals of Δt . In our implementation, we set an arbitrary lower threshold for Δt as 2 ms and an upper threshold of $t_i/50$, where t_i is the spline duration. For splines with durations lower than 2 ms, we skip adding the optimization variables as well as the corresponding objective terms and constraints to our optimization.

The ZMP-based inequality constraints (25) in base TO also require discretization of the planning horizon. The sampled time, Δt , in seconds, was obtained as $\Delta t = \min(0.085, \Delta t_{\text{event}})$, where Δt_{event} is the time until the next touchdown/liftoff event caused by one or more legs.

C. Analysis

We use the mechanical cost of transport (COT_m), a dimensionless quantity, to compare the energy efficiencies of different gaits during locomotion. For legged robots, we follow the definition as in [5], where it is the ratio of the power inputs from the motors to the power equivalence of the locomotion speed, given as,

$$\text{COT}_m = \frac{\frac{1}{N} \sum_{k=1}^N \sum_{j=1}^4 \sum_{i=1}^4 \max(\tau_{ijk} u_{ijk}, 0)}{mgv_{\text{avg}}}, \quad (28)$$

where τ_{ijk} is the produced torque and u_{ijk} the angular velocity of the joint i of leg j at the time instant k (of a total N samples). The variables m , g and v_{avg} denote the mass of the base, the gravitational acceleration and the average velocity of the base in \mathcal{W} -frame.

VI. RESULTS AND DISCUSSIONS

A. Gaits

The proposed motion planner described in Sec. III and IV is greatly versatile to multiple gaits, whose timings and sequences of liftoff and touchdown events are predefined. The robot is able to execute pure driving, hybrid crawl, hybrid trot,

hybrid pace and hybrid running trot, as demonstrated in the accompanying video⁴.

The pure driving is a reliable gait for straight-line motions as all four legs are on the ground, providing a larger support polygon. While turning, however, the legs can get overextended from the base. For an optimization horizon of 2 s, the maximum angle the robot can turn while driving in this time, without falling, is 25° .

While executing a hybrid crawling gait (see Fig. 5), at least three legs are in stance at any given time. The support polygons are either triangles or quadrilaterals, which makes crawl the ideal gait for applications where stability and accurate placement of the feet have a higher priorities than speed [51].

Trotting is characterized by two diagonal legs swinging together and was recently shown to be the most effective gait in terms of cost of transport (COT) for conventional legged robots [52]. Despite the stability concerns due to support polygons being reduced to support lines, the robot retains balance due to our zero-base-pitch w.r.t. ground assumption. The trot is generally known to be quite stable as the CoM does not shift too much laterally [53]. Therefore, hybrid trot is used as a default gait in our work.

While executing a pacing gait, the two legs on the same side are swinging simultaneously, and hence the support lines switch between the lines of the two lateral feet. To execute a pace, the base TO generates significant accelerations in the lateral directions such that the ZMP stays in these support lines, while the CoM tries to sway as little as possible, since we minimize deviations from a path regularizer.

Running (or flying) trot is an extension of trotting with full flight phases. This means, there is significant motion of the base CoM in the direction perpendicular to the terrain. Therefore, the z component of the CoM trajectory needs to be optimized as well, making the ZMP inequality constraints, and hence the base TO nonlinear in its optimization variables.

B. Solver Times

As shown in Table II, the wheel and base optimizations are solved in the order of milliseconds, allowing the optimizations to update frequently. Thanks to these high update rates, the robot is able to cope with unforeseen disturbances and modeling errors.

⁴<https://youtu.be/ukY0vyM-yfY>

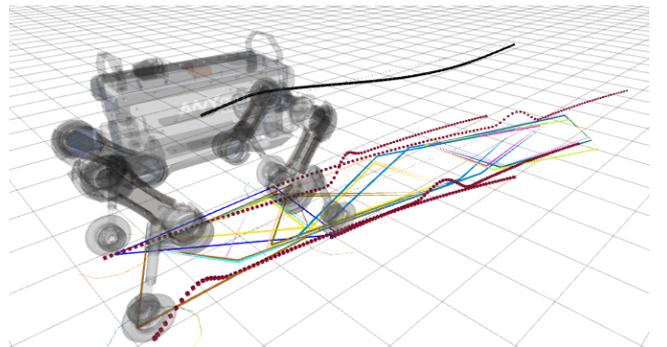


Fig. 5. Simulation of ANYmal on wheels while hybrid-crawling in a curve. The planned trajectories for the wheels (red, dotted) and the base CoM (black, solid) are shown. The support polygons (multi-coloured) generated from the foot TO, are sent to the base TO to provide constraints on the ZMP.

TABLE II
PLANNING HORIZON AND THE AVERAGE OPTIMIZATION TIMES FOR DIFFERENT GAITS. THE OPTIMIZATION VARIABLES AND NUMBER OF CONSTRAINTS ARE EXPRESSED AS A RANGE, AS <MIN> / <MAX>.

Gait	Planning Horizon / (s)	Proposed Feet TO			Solver Time* / (ms)	Proposed Base TO			Solver Time / (ms)
		#optimization variables	#equality constraints	#inequality constraints		#optimization variables	#equality constraints	#inequality constraints	
Driving	1.7	5	2	200	0.14	232	116	528	6.93
Hybrid Crawl	2.0	43 / 61	31 / 43	240 / 452	0.81	243 / 264	112 / 124	428 / 470	14.83
Hybrid Trot	0.85	43 / 61	31 / 43	124 / 192	0.47	109 / 127	36 / 40	252 / 280	2.40
Hybrid Pace	0.95	43 / 61	31 / 43	140 / 200	0.42	88 / 119	28 / 36	196 / 252	1.88
Hybrid Running Trot	0.64	43 / 61	31 / 43	116 / 192	0.58	59 / 82	24 / 36	76 / 104	5.77

*for one foot.

The solver times for both the feet and base TOs seem to depend primarily on the number of optimization variables and the time horizon. The number of optimization variables for the wheel TO, n_s , varied depending on the spline sequence (see Fig. 3) in that optimization. In our feet TO formulation, all hybrid gaits had the same range of values for n_s , making comparisons between solver times and planning horizon easier. Longer planning horizons had higher samples of the trajectories for the inequality constraints (kinematic bounds), which made solvers take more time.

The planning horizon dictates then number of optimization variables for base TO through the number of support polygons. A higher planning horizon has more support polygons, and hence more connecting splines. This is associated with more variables to optimize for (see Section IV-A), which in turn increases the solver times. The base TO while hybrid running trot, however, takes more time to optimize than hybrid trot, despite having lower optimization variables. This is because, as mentioned earlier, the hybrid running trot makes the base TO nonlinear. Thanks to the custom SQP algorithm, the solver times for hybrid trot are still comparable to the other gaits and can be used reliably on the robot.

Our TO framework has solver times comparable with that of ANYmal without wheels, mentioned in [32], [34], for the gaits described above. Moreover, our algorithm performs significantly better with regards to solver times than [26] where the feet and base trajectories are optimized simultaneously, which took 20 ms for the same hybrid gaits.

C. Speed and Cost of Transportation

The maximum heading speeds for the base (obtained from an external source) for some hybrid gaits are mentioned in Table III. Our hybrid gaits were much faster than the same gaits employed in ANYmal without wheels.

The energy efficiency of various gaits on ANYmal are quantified using mechanical cost of transport (28) and compared in Table IV. The COT_m for our default gait, hybrid trot, was lower than the conventional trotting gait by 42% and by 9% for skating motions [5]. Although our hybrid trot has a COT_m twice as much as pure driving, it is impressive that such dynamic motions can be performed at the same speed of pure driving.

TABLE III
MAXIMUM HEADING SPEEDS OF THE BASE, v_{max} , FOR DIFFERENT GAITS

Gaits	v_{max} for conventional gaits / (m/s)	v_{max} for hybrid versions / (m/s)
Crawl	0.15	1.0
Trot	0.5	2.0
Pace	0.4	2.0
Running Trot	0.4	1.0

TABLE IV
COMPARISON OF MECHANICAL COT BETWEEN MULTIPLE GAITS

Gait	Power Consumed / (W)	v_{avg} / (m/s)	COT_m	Reference
Crawling	18.35	0.06	1.0	[5]
Trot	50.68	0.29	0.6	[5]
Pure Driving	63.64	2.00	0.1	[4]
Hybrid Trot	156.00	2.00	0.2	-

The most ideal situation, for reducing the COT_m would be switching between driving and a hybrid gait depending on the need. The switching method, explained in Section III-H, was not very reliable in simulation, and hence was not used on the robot. This could be because the algorithm does not specify the order in which the legs must be lifted. Therefore, a leg which is already in the verge of overextension can get overextended, if not lifted first, before others. Therefore, for now, we manually switch between driving and hybrid trotting.

D. Reactive Behavior

Although our motion planning framework relies on a flat-terrain assumption, the robot is versatile enough to traverse over several terrains like shallow steps, gravel, mud and puddles. It traversed over a block of height 8 cm (13% of its leg length) blindly, which was more than the radius of the wheels (5 cm).

When ANYmal encounters uneven terrains while locomoting blindly, the height variations on the terrain appear as disturbances. ANYmal quickly recovers balance due to the fast update rates of the TOs and motion controller. A new



Fig. 6. ANYmal on wheels reacting to external force disturbances from a human. The inverted pendulum term in foothold predictions helps to guide the placement of the feet to recover balance.

motion plan is created as soon as it encounters an obstacle on the terrain.

ANYmal on wheels is significantly robust against unexpected force disturbances on the base, as shown in Fig. 6. When an external force is acting on the base without any external velocity commands, the feet TO predicts the next foothold to obtain a stable configuration, using the inverted pendulum model (16). Then the base trajectory adapts such that the ZMP moves appropriately to the new support polygon.

E. Real-World Application

We deployed ANYmal on wheels at the Tunnel Circuit of the DARPA Subterranean Challenge [54] in Pittsburgh, USA. The objective of the challenge was to navigate through a swamp terrain and explore the underground mine.

As depicted in the lower images of Fig. 7, the terrain consisted of hilly, bumpy, and muddy terrain and in some parts of the mine, the robot needed to cross puddles. Throughout the challenge, the robot traversed these challenging terrains with a hybrid trot. In the first run of the challenge, the wheeled version of ANYmal managed to traverse 70 m without major issues. In the end, however, one of the wheels started slipping on the muddy terrain. As can be seen in the accompanying video, the robot manages to balance after the first slip because of our implementation of the inverted pendulum model in (16). The mechanical design was improved after the first run by adding a chain around the wheels to increase the friction coefficient while traversing the mud (see Fig. 7).

Fig. 8 and 9 show the measured and desired trajectories of the CoM and wheels, respectively, for a few meters of the subsequent run. ANYmal on wheels uses a hybrid trot and manages to continuously explore for more than 100 m. In Fig. 9, we can see that the robot executes sufficiently smooth motions even on very unpredictable terrains as the commanded accelerations are minimal and devoid of large jumps, except at the times of touchdown.

Due to the time limitation of the challenge, the speed of mobile platforms becomes an important factor. Most of the wheeled platforms from the other competing teams were faster than our traditional legged robot by more than a factor of two. The upcoming Urban Circuit of the Subterranean Challenge includes stairs and other challenging obstacles. Therefore, we

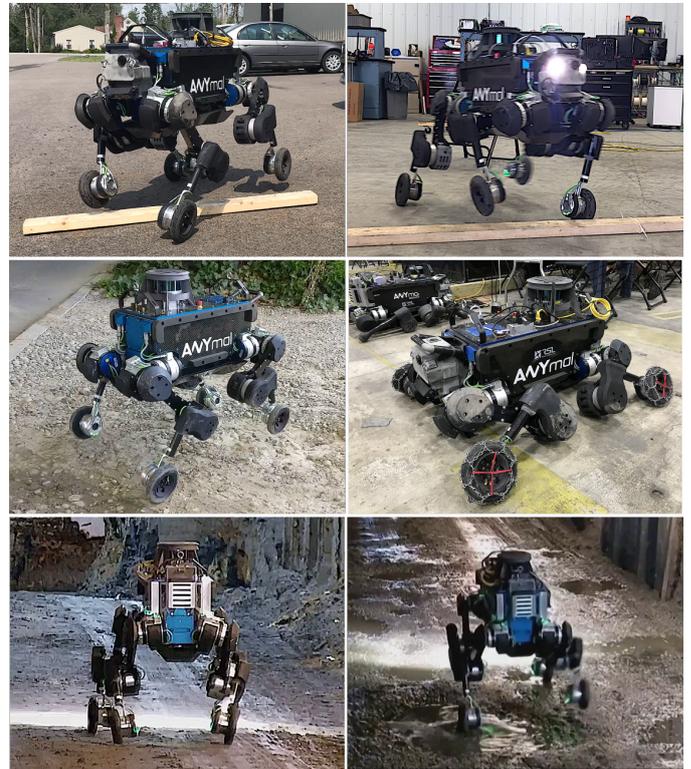


Fig. 7. ANYmal on wheels traversing over a wooden plank (top images), on rough terrains (left middle image) and muddy underground environments at the DARPA Subterranean Challenge (lower images). The wheels are equipped with chains to traverse the slippery muddy terrain (right middle image).

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

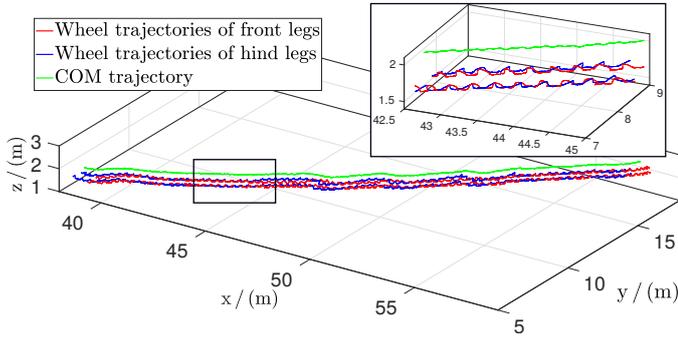


Fig. 8. Measured base and feet trajectories of ANYmal at the DARPA Subterranean Challenge. ANYmal executes a hybrid trot on wet and muddy underground environments. The three-dimensional plot shows the feet trajectories of the front legs (red line), the feet trajectories of the hind legs (blue line), and the CoM trajectory (green line) w.r.t. the inertial frame, which is initialized at the beginning of the run. Moreover, the robot managed to continuously explore the mine for more than 100 m.

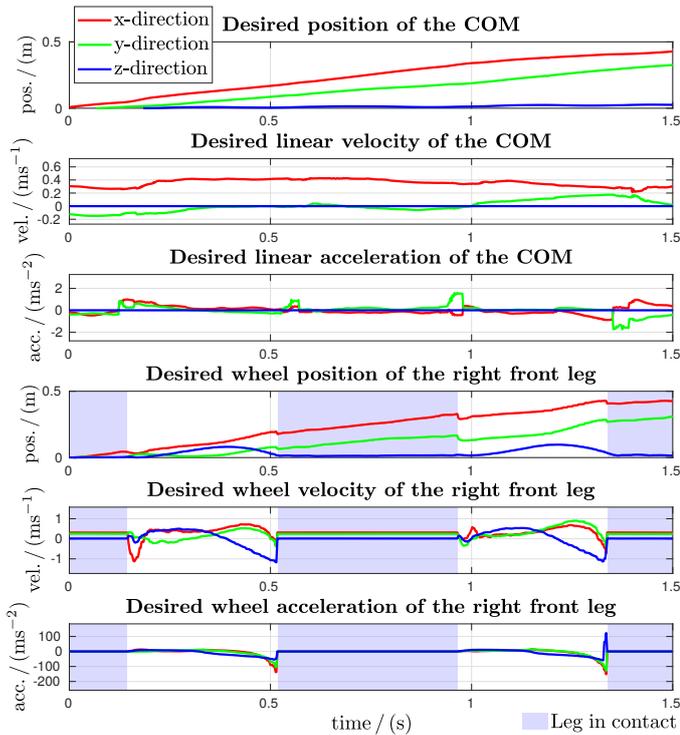


Fig. 9. Desired CoM and feet trajectories of ANYmal at the DARPA Subterranean Challenge while hybrid trotting. The plots show the desired motions for approximately two stride durations of the run shown in Fig. 8.

VII. CONCLUSIONS AND FUTURE WORK

This work presents an online TO framework for generating hybrid walking-driving motions on a wheeled quadrupedal robot. The optimization problem is broken down into feet and base trajectory generation, with the aim of reducing solver times. The independent feet and base TOs are synchronized to generate feasible motions by time sampling the prior generated feet trajectories which generate the support polygons for the ZMP inequality constraint of the base TO. The presented algorithm makes the locomotion planning for high dimensional

wheeled-legged robots, like ANYmal, more tractable, enables us to solve the problem in real-time on-board in a MPC fashion, and increases the robustness in the robot's locomotion against unforeseen disturbances.

To the best of our knowledge, this is the first time that a hybrid walking-driving robot is deployed for real-world missions at one of the biggest robotics competition. To improve the reliability of our newly developed platform, the hardware maturity needs to be increased, e.g., improve the grip of the wheels. Additionally, an algorithmic approach with a higher-level intelligence to judge when to switch between a pure driving gait and a hybrid walking-driving gait can be beneficial. In the future, we seek to integrate perception into the motion planning framework for stair climbing and other applications requiring more careful and precise foot placements.

We hope that our work serves as an inspiration for the existing quadrupedal systems (such as Spot Mini, Cheetah, HyQ, Likago and Ghost robots, which share similar designs with ANYmal in terms of degrees of freedom and actuator placement in the legs) to employ hybrid motions with additional wheels, without the need for a separate steering mechanism.

ACKNOWLEDGMENTS

The authors would like to thank Dario Bellicoso and Fabian Jenelten for their help on the code. Additionally, a special thanks to all members of the CERBERUS team at DARPA Subterranean Challenge consisting of University of Nevada, Reno, ETH Zürich, Sierra Nevada Corporation, University of California, Berkeley, and Flyability.

APPENDIX A

ANYMAL'S ABILITY TO MAKE TURNS WITHOUT STEERING THE WHEELS

In Fig. 10, the pose of ANYmal's base is represented using a body-fixed coordinate system at its CoM, B , associated with \mathcal{B} -frame with x_B along its heading direction. Let ω_{des} be the angular velocity of the base in an inertial \mathcal{C} -frame, provided externally, for instance, using a joystick. Let the instantaneous center of rotation (ICR) of the ANYmal's base be N .

Consider one wheel and let F be a point that represents the center of wheel's axle. The velocity of F , \mathbf{v}_F , is known to be pointing in the heading direction of the base, perpendicular to the axle of the wheel as shown. Let F' be a point fixed to the coordinate frame of the base, that happens to momentarily coincide with F . Since F' is a part of the base, the velocity of this point, $\mathbf{v}_{F'}$ is perpendicular to the line connecting the ICR of the base and F' , $\mathbf{r}_{NF'}$, with $\mathbf{v}_{F'} = \omega_{\text{des}} \times \mathbf{r}_{NF'}$.

Given the magnitude and directions of both these velocity vectors, the relative velocity between F and F' , $\mathbf{v}_{F/F'}$ is given by $\mathbf{v}_{F/F'} = \mathbf{v}_F - \mathbf{v}_{F'}$.

Since the wheels are not fixed to the body, the wheels move to a configuration as shown in Fig. 11 due to this relative velocity, thereby allowing turns without having to steer the wheels. ANYmal cannot continue turning this way for long as the legs get overextended eventually. To continue motion, the robot must start stepping to get back to stable configurations to avoid fall.

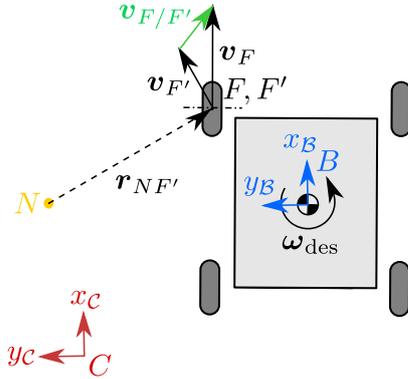


Fig. 10. Initial configuration of ANYmal while turning.

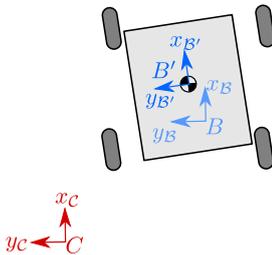


Fig. 11. Final configuration of ANYmal while turning. The coordinate system of the base has moved from $B\{x_B, y_B, z_B\}$ to $B'\{x_{B'}, y_{B'}, z_{B'}\}$.

APPENDIX B

REPRESENTING THE OBJECTIVE TO AVOID EXTENSION OF LEGS IN STANCE IN QP FORM

Expanding ${}^{\mathcal{P}'}\mathbf{r}_{\mathcal{P}'D'}$ at time instant t_k ,

$$\begin{aligned} {}^{\mathcal{P}'}\mathbf{r}_{\mathcal{P}'D'}(t_k) &= \mathbf{R}_{\mathcal{P}'\mathcal{P}}(t_k) {}^{\mathcal{P}}\mathbf{r}_{\mathcal{P}'D'}(t_k) \\ &= \mathbf{R}_{\mathcal{P}'\mathcal{P}}(t_k) ({}^{\mathcal{P}}\mathbf{r}_{\mathcal{P}D'}(t_k) - {}^{\mathcal{P}}\mathbf{r}_{\mathcal{P}P'}(t_k)) \\ &= \mathbf{R}_{\mathcal{P}'\mathcal{P}}(t_k) ({}^{\mathcal{P}}\mathbf{r}_{\mathcal{P}D'}(t_k) - \mathbf{T}_{\text{st}}(\omega_{\text{des}}^z, t_k) \zeta_{\text{st}}) \\ &= \underbrace{\mathbf{R}_{\mathcal{P}'\mathcal{P}}(t_k) {}^{\mathcal{P}}\mathbf{r}_{\mathcal{P}D'}(t_k)}_{\delta} \\ &\quad + \underbrace{(-\mathbf{R}_{\mathcal{P}'\mathcal{P}}(t_k) \mathbf{T}_{\text{st}}(\omega_{\text{des}}^z, t_k))}_{\Gamma} \zeta_{\text{st}}, \end{aligned} \quad (29)$$

with $\Gamma \in \mathbb{R}^{3 \times 5}$ and $\delta \in \mathbb{R}^3$.

We minimize the x component of this position vector, ${}^{\mathcal{P}'}\mathbf{r}_{\mathcal{P}'D'}^x(t_k) = \Gamma^x \zeta_{\text{st}} + \delta^x$, where $\Gamma^x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \Gamma$ and $\delta^x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \delta$. The objective function is written as,

$$\begin{aligned} &\sum_{k=1}^N w_{\text{def}} \|{}^{\mathcal{P}'}\mathbf{r}_{\mathcal{P}'D'}^x(t_k)\|^2 \Delta t \\ &= \frac{1}{2} \zeta_{\text{st}}^T \underbrace{\left(\sum_{k=1}^N 2w_{\text{def}} (\Gamma^x)^T \Gamma^x \Delta t \right)}_{\mathbf{Q}_{\text{def}}} \zeta_{\text{st}} \\ &\quad + \underbrace{\left(\sum_{k=1}^N 2w_{\text{def}} \delta^x \Gamma^x \Delta t \right)}_{\mathbf{c}_{\text{def}}^T} \zeta_{\text{st}}. \end{aligned} \quad (30)$$

APPENDIX C

OBJECTIVES FOR MINIMIZING DEVIATION FROM PREVIOUS SOLUTION

Deviations in position are minimized as in (11) over the entire optimization horizon, T . This horizon of one gait duration contains both stance and spline segments, each contributing to the objective function. For instance, consider a swing segment that starts at time t_1 and ends at t_2 . Equation (11) for this spline would then appear as,

$$\sum_{k=1}^n \left\| \mathbf{T}_{\text{sw}}(t_k) \zeta_{\text{sw}} - {}^{\mathcal{P}}\mathbf{r}_{\mathcal{P}P'}^{i-1}(t_k + t') \right\|_{\mathbf{W}_{\text{pre,sw}}^{\text{pos}}}^2 \Delta t, \quad \forall t_k \in [t_1, t_2], \quad (31)$$

where $\Delta t = (t_2 - t_1)/n$. Expanding this, we get the cost function hessian and the linear term for this swing segment, as,

$$\begin{aligned} \mathbf{Q}_{\text{pre,sw}}^{\text{pos}} &= \sum_{k=1}^n 2\mathbf{T}_{\text{sw}}^T(t_k) \mathbf{W}_{\text{pre,sw}}^{\text{pos}} \mathbf{T}_{\text{sw}}(t_k) \Delta t, \\ \mathbf{c}_{\text{pre,sw}}^{\text{pos}} &= \sum_{k=1}^n 2\mathbf{T}_{\text{sw}}^T(t_k) \mathbf{W}_{\text{pre,sw}}^{\text{pos}} {}^{\mathcal{P}}\mathbf{r}_{\mathcal{P}P'}^{i-1}(t_k + t') \Delta t. \end{aligned} \quad (32)$$

To obtain (11), we add up the cost function Hessians and linear terms for each spline in the sequence.

We also minimize the deviations in velocity and accelerations over the optimization horizon, formulated as,

$$\sum_{k=1}^N \left\| \mathcal{P} \mathbf{v}_{P'}^i(t_k) - \mathcal{P} \mathbf{v}_{P'}^{i-1}(t_k + t') \right\|_{\mathbf{W}_{\text{pre}}^{\text{vel}}}^2 \Delta t, \quad \forall t_k \in [0, T], \quad (33)$$

$$\sum_{k=1}^N \left\| \mathcal{P} \mathbf{a}_{P'}^i(t_k) - \mathcal{P} \mathbf{a}_{P'}^{i-1}(t_k + t') \right\|_{\mathbf{W}_{\text{pre}}^{\text{acc}}}^2 \Delta t, \quad \forall t_k \in [0, T]. \quad (34)$$

APPENDIX D

REPRESENTING THE INEQUALITY CONSTRAINTS TERM IN QP FORM

Expanding $\mathcal{P}' \mathbf{r}_{P'H'}$,

$$\begin{aligned} \mathcal{P}' \mathbf{r}_{P'H'}(t_k) &= \mathbf{R}_{\mathcal{P}'\mathcal{P}}(t_k) \mathcal{P} \mathbf{r}_{P'H'}(t_k) \\ &= \mathbf{R}_{\mathcal{P}'\mathcal{P}}(t_k) (\mathcal{P} \mathbf{r}_{B'H'}(t_k) + \mathcal{P} \mathbf{r}_{P'B'}(t_k)) \\ &= \mathbf{R}_{\mathcal{P}'\mathcal{P}}(t_k) \\ &\quad (\mathbf{R}_{\mathcal{P}B'}(t_k) \mathcal{B}' \mathbf{r}_{B'H'}(t_k) + \mathcal{P} \mathbf{r}_{P'B'}(t_k)), \end{aligned} \quad (35)$$

where $\mathcal{B}' \mathbf{r}_{BH}(t_k) = \mathcal{B}' \mathbf{r}_{B'H'}(t_k) \quad \forall t_k \in [0, T]$, as the hip does not move w.r.t the base. For brevity, hereafter, we write vectors and matrices without t_k , e.g., $\mathbf{R}_{\mathcal{P}'\mathcal{P}}(t_k)$ as $\mathbf{R}_{\mathcal{P}'\mathcal{P}}$. Therefore,

$$\begin{aligned} \mathcal{P}' \mathbf{r}_{P'H'} &= \mathbf{R}_{\mathcal{P}'\mathcal{P}} (\mathbf{R}_{\mathcal{P}B'} \mathcal{B}' \mathbf{r}_{BH} + \mathcal{P} \mathbf{r}_{P'B'}) \\ &= \mathbf{R}_{\mathcal{P}'\mathcal{P}} \mathbf{R}_{\mathcal{P}B'} \mathcal{B}' \mathbf{r}_{BH} + \mathbf{R}_{\mathcal{P}'\mathcal{P}} \mathcal{P} \mathbf{r}_{P'B'} \\ &= \mathbf{R}_{\mathcal{P}'\mathcal{B}'} \mathcal{B}' \mathbf{r}_{BH} + \mathbf{R}_{\mathcal{P}'\mathcal{P}} \mathcal{P} \mathbf{r}_{P'B'} \end{aligned} \quad (36)$$

Assuming zero base pitch and roll angles w.r.t. terrain plane, we get $\mathbf{R}_{\mathcal{P}'\mathcal{B}'} = \mathbf{R}_{\mathcal{P}B'} = \mathbb{I}_{3 \times 3}$. Using this,

$$\begin{aligned} \mathcal{P}' \mathbf{r}_{P'H'} &= \mathbf{R}_{\mathcal{P}B'} \mathcal{B}' \mathbf{r}_{BH} + \mathbf{R}_{\mathcal{P}'\mathcal{P}} \mathcal{P} \mathbf{r}_{P'B'} \\ &= \mathbf{R}_{\mathcal{P}B'} \mathcal{B}' \mathbf{r}_{BH} + \mathbf{R}_{\mathcal{P}'\mathcal{P}} (\mathcal{P} \mathbf{r}_{PB'} - \mathcal{P} \mathbf{r}_{PP'}) \end{aligned} \quad (37)$$

Assuming that at time t_k , the spline corresponds to a swing, we substitute $\mathcal{P} \mathbf{r}_{PP'}(t_k) = \mathbf{T}_{\text{sw}}(t_k) \boldsymbol{\zeta}_{\text{sw}}$.

$$\begin{aligned} \mathcal{P}' \mathbf{r}_{P'H'} &= \underbrace{\mathbf{R}_{\mathcal{P}B'} \mathcal{B}' \mathbf{r}_{BH} + \mathbf{R}_{\mathcal{P}'\mathcal{P}} \mathcal{P} \mathbf{r}_{PB'}}_{\boldsymbol{\mu}} \\ &\quad + \underbrace{(-\mathbf{R}_{\mathcal{P}'\mathcal{P}} \mathbf{T}_{\text{sw}}(t_k))}_{\boldsymbol{\Lambda}} \boldsymbol{\zeta}_{\text{sw}}, \end{aligned} \quad (38)$$

with $\boldsymbol{\Lambda} \in \mathbb{R}^{3 \times 18}$ and $\boldsymbol{\mu} \in \mathbb{R}^3$.

Consider the inequality constraint, $|\mathcal{P}' \mathbf{r}_{P'H'}^x| < x_{\text{offset}}$. This can be decomposed into two inequalities:

$$\boldsymbol{\Lambda}^x \boldsymbol{\zeta} + \boldsymbol{\mu}^x < x_{\text{offset}} \implies \underbrace{\boldsymbol{\Lambda}^x \boldsymbol{\zeta}}_{\mathbf{D}} < \underbrace{x_{\text{offset}} - \boldsymbol{\mu}^x}_{\mathbf{e}}, \quad (39)$$

and,

$$-x_{\text{offset}} < \boldsymbol{\Lambda}^x \boldsymbol{\zeta} + \boldsymbol{\mu}^x \implies \underbrace{-\boldsymbol{\Lambda}^x \boldsymbol{\zeta}}_{\mathbf{D}} < \underbrace{x_{\text{offset}} + \boldsymbol{\mu}^x}_{\mathbf{e}}, \quad (40)$$

where, $\boldsymbol{\Lambda}^x$ is the first row of $\boldsymbol{\Lambda}$ and $\boldsymbol{\mu}^x$, the first element of $\boldsymbol{\mu}$, obtained as,

$$\boldsymbol{\Lambda}^x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \boldsymbol{\Lambda}, \quad \boldsymbol{\mu}^x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \boldsymbol{\mu}. \quad (41)$$

Inequalities (39) and (40) can be written for the y coordinate as well. Therefore, for every time step t_k , we have four

inequality constraints, corresponding to each side of the rectangle foot bound. These inequalities can be extended for stance splines with the substitution $\mathcal{P} \mathbf{r}_{PP'}(t_k) = \mathbf{T}_{\text{st}}(\omega_{\text{des}}^z, t_k) \boldsymbol{\zeta}_{\text{st}}$.

APPENDIX E

SHIFTING FROM DRIVING TO WALKING

Let $\mathbf{v}_{\text{des}} = [v_{\text{des}}^x \ v_{\text{des}}^y \ 0]^T$ and $\boldsymbol{\omega}_{\text{des}} = [0 \ 0 \ \omega_{\text{des}}^z]^T$ be the reference velocities of the base. From Appendix A, we know that the relative velocity between a foot and the base ($\mathbf{v}_{F/F'}$) provides the possibility of small turns. Following the notations in Fig. 10, we write all vectors expressed in the coordinate system of the base, $B\{x_B, y_B, z_B\}$, as

$$\begin{aligned} \mathbf{v}_{F/F'} &= \mathbf{v}_F - \mathbf{v}_{F'}, \\ &= \mathbf{v}_{\text{des}} - \boldsymbol{\omega}_{\text{des}} \times \mathbf{r}_{NF'}, \\ &= \mathbf{v}_{\text{des}} - \boldsymbol{\omega}_{\text{des}} \times (\mathbf{r}_{BF'} - \mathbf{r}_{BN}), \end{aligned} \quad (42)$$

where the position vector $\mathbf{r}_{BF'} = \mathbf{r}_{\text{mea}} = [x_{\text{mea}} \ y_{\text{mea}} \ 0]^T$ is the measured position of the feet from the CoM of base. From $\mathbf{v}_{\text{des}} \perp \mathbf{r}_{BN}$, we can deduce that

$$\mathbf{r}_{BN} = \begin{bmatrix} v_{\text{des}}^y / \omega_{\text{des}}^z \\ -v_{\text{des}}^x / \omega_{\text{des}}^z \\ 0 \end{bmatrix} \quad (43)$$

Substituting this in (42), we get,

$$\mathbf{v}_{F/F'} = \begin{bmatrix} y_{\text{mea}} \omega_{\text{des}}^z \\ -x_{\text{mea}} \omega_{\text{des}}^z \\ 0 \end{bmatrix} \quad (44)$$

For an optimization horizon of T , the foot moves by $\mathbf{v}_{F/F'} T$ from the measured position $\mathbf{r}_{BF'}$. Note that this displacement does not depend on the linear heading velocity, \mathbf{v}_{des} .

If this predicted foot position stays within a circle of radius l_{max} centered around the default foot position from the standing configuration, $\mathbf{r}_{\text{def}} = [x_{\text{def}} \ y_{\text{def}} \ 0]^T$, then the robot can just turn by driving. This can be formulated as,

$$\left\| (\mathbf{r}_{\text{mea}} + \mathbf{v}_{F/F'} T) - \mathbf{r}_{\text{def}} \right\|^2 \leq l_{\text{max}}^2. \quad (45)$$

On simplification,

$$\begin{aligned} (x_{\text{mea}} - x_{\text{def}} + y_{\text{mea}} \omega_{\text{des}}^z T)^2 \\ + (y_{\text{mea}} - y_{\text{def}} - x_{\text{mea}} \omega_{\text{des}}^z T)^2 \leq l_{\text{max}}^2. \end{aligned} \quad (46)$$

If the condition is violated even for a foot, we switch to walking.

For a specific case of $\mathbf{r}_{\text{mea}} = \mathbf{r}_{\text{def}}$, the condition for walking can be formulated as,

$$|\omega_{\text{des}}^z| \geq \frac{l_{\text{max}}}{T \sqrt{x_{\text{def}}^2 + y_{\text{def}}^2}}, \quad (47)$$

clearly showing that for larger reference angular velocities, the robot must walk.

REFERENCES

- [1] M. Xiangrui, W. Shuo, C. Zhiqiang, and Z. Leijie, "A review of quadruped robots and environment perception," *2016 35th Chinese Control Conference (CCC)*, pp. 6350–6356, 2016.
- [2] M. Hutter, R. Diethelm, S. Bachmann, P. Fankhauser, C. Gehring, V. Tsounis, A. Lauber, F. Guenther, M. Bjelonic, L. Isler, H. Kolvenbach, K. Meyer, and M. Hoepflinger, "Towards a generic solution for inspection of industrial sites." ETH Zurich, 2017-09, 11th Conference on Field and Service Robotics (FSR) 2017; Conference Location: Zurich, Switzerland; Conference Date: September 12-15, 2017.
- [3] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 172988141983959, 2019. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/1729881419839596>
- [4] M. Bjelonic, 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." [Online]. Available: <https://arxiv.org/pdf/1809.03557.pdf>
- [5] M. Bjelonic, C. D. Bellicoso, M. E. Tiryaki, and M. Hutter, "Skating with a force controlled quadrupedal robot," 2018, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018)*; Conference Location: Madrid, Spain; Conference Date: October 1-5, 2018; RSL; snf; dfab.
- [6] G. Endo and S. Hirose, "Study on roller-walker - energy efficiency of roller-walk," *2011 IEEE International Conference on Robotics and Automation*, pp. 5050–5055, 2011.
- [7] W. Reid, F. J. Perez-Grau, A. H. Goktogan, and S. Sukkarieh, "Actively articulated suspension for a wheel-on-leg rover operating on a Martian analog surface," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 5596–5602, 2016.
- [8] K. Iagnemma, A. Rzepniewski, S. Dubowsky, and P. Schenker, "Control of robotic vehicles with actively articulated suspensions in rough terrain," *Autonomous Robots*, vol. 14, no. 1, pp. 5–16, Jan 2003. [Online]. Available: <https://doi.org/10.1023/A:1020962718637>
- [9] S. YLÄNEN and A. Halme, "Further development and testing of the hybrid locomotion of workpartner robot," 01 2002.
- [10] M. Takahaashi, K. Yoneda, and S. Hirose, "Rough terrain locomotion of a leg-wheel hybrid quadruped robot," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 1090–1095.
- [11] P. Robuffo Giordano, M. Fuchs, A. Albu-Schäffer, and G. Hirzinger, "On the kinematic modeling and control of a mobile platform equipped with steering wheels and movable legs," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4080–4087, 2009.
- [12] A. Dietrich, T. Wimböck, A. Albu-Schäffer, and G. Hirzinger, "Singularity avoidance for nonholonomic, omnidirectional wheeled mobile platforms with variable footprint," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 6136–6142, 2011.
- [13] M. Fuchs, C. Borst, P. R. Giordano, A. Baumann, E. Kraemer, J. Langwald, R. Gruber, N. Seitz, G. Plank, K. Kunze, R. Burger, F. Schmidt, T. Wimboeck, and G. Hirzinger, "Rollin' Justin - Design considerations and realization of a mobile platform for a humanoid upper body," *Proceedings - IEEE International Conference on Robotics and Automation*, no. May 2014, pp. 4131–4137, 2009.
- [14] F. Cordes, C. Oekermann, A. Babu, D. Kuehn, T. Stark, and F. Kirchner, "An Active Suspension System for a Planetary Rover," *ISAIRAS*, no. February 2017, 2014.
- [15] M. Gifthalder, F. Farshidian, T. Sandy, L. Stadelmann, and J. Buchli, "Efficient kinematic planning for mobile manipulators with non-holonomic constraints using optimal control," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3411–3417, 2017.
- [16] J. P. Laumond, S. Sekhavat, and F. Lamiroux, *Guidelines in nonholonomic motion planning for mobile robots*, 2006.
- [17] M. Wada and H. H. Asada, "Design of a Holonomic Omnidirectional Vehicle Using a Reconfigurable Footprint Mechanism and Its Application to a Wheelchair," *Journal of the Robotics Society of Japan*, vol. 16, no. 6, pp. 816–823, 2012.
- [18] A. Suzumura and Y. Fujimoto, "Real-time motion generation and control systems for high wheel-legged robot mobility," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3648–3659, 2014.
- [19] C. Grand, F. Benamar, and F. Plumet, "Motion kinematics analysis of wheeled-legged rover over 3D surface with posture adaptation," *Mechanism and Machine Theory*, vol. 45, no. 3, pp. 477–495, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0094114X09001918>
- [20] K. K. Hauser, T. Bretl, J.-C. Latombe, and B. Wilcox, "Motion planning for a six-legged lunar robot," vol. 47, 01 2006, pp. 301–316.
- [21] M. Olinski and J. Ziemba, *Hybrid Quadruped Robot à Mechanical Design and Gait Modelling*, 01 2014, vol. 17, pp. 183–190.
- [22] T. Klamt, D. Rodriguez, M. Schwarz, C. Lenz, D. Pavlichenko, D. Droschel, and S. Behnke, "Supervised Autonomous Locomotion and Manipulation for Disaster Response with a Centaur-like Robot," no. October, 2018. [Online]. Available: <http://arxiv.org/abs/1809.06802>
- [23] A. Laurenzi, E. M. Hoffman, and N. G. Tsagarakis, "Quadrupedal walking motion and footstep placement through Linear Model Predictive Control," pp. 2267–2273, 2018.
- [24] T. Klamt and S. Behnke, "Anytime hybrid driving-stepping locomotion planning," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Sept, pp. 4444–4451, 2017.
- [25] M. Schwarz, T. Rodehutsors, D. Droschel, M. Beul, M. Schreiber, N. Araslanov, I. Ivanov, C. Lenz, J. Razlaw, S. Schüller, D. Schwarz, A. Topalidou-Kyniazopoulou, and S. Behnke, "NimRo Rescue: Solving Disaster-response Tasks with the Mobile Manipulation Robot Momaro," *Journal of Field Robotics*, vol. 34, no. 2, pp. 400–425, 2017.
- [26] Y. de Viragh, M. Bjelonic, C. D. Bellicoso, F. Jenelten, and M. Hutter, "Trajectory Optimization for Wheeled-Legged Quadrupedal Robots Using Linearized ZMP Constraints," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1633–1640, 2019.
- [27] M. Geilinger, R. Poranne, R. Desai, B. Thomaszewski, and S. Coros, "Skaterbots," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–12, 2018. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3197517.3201368>
- [28] B. Dynamics, "Introducing handle," <https://www.youtube.com/watch?v=-7xvqQeoA8c>, Feb 2017.
- [29] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 38–44.
- [30] 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*, no. 99, pp. 1–1, 2018. [Online]. Available: <http://ieeexplore.ieee.org/document/8283570/>
- [31] C. Gehring, S. Coros, M. Hutler, C. D. Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. Hoepflinger, and R. Siegwart, "Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot," *IEEE Robotics Automation Magazine*, vol. 23, no. 1, pp. 34–43, March 2016.
- [32] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic Locomotion through Online Nonlinear Motion Optimization for Quadrupedal Robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1–1, 2018. [Online]. Available: <http://ieeexplore.ieee.org/document/8260889/>
- [33] A. W. Winkler, "Optimization-based motion planning for legged robots," p. 100, 2018. [Online]. Available: <https://doi.org/10.3929/ethz-b-000272432>
- [34] C. Dario Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Sept, pp. 3359–3365, 2017.
- [35] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," 01 2007, pp. 200 – 207.
- [36] M. Raibert, *Legged Robots that Balance*, ser. Artificial Intelligence. MIT Press, 1986. [Online]. Available: <https://books.google.nl/books?id=EXRiBnQ37RwC>
- [37] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, Sep 1983. [Online]. Available: <https://doi.org/10.1007/BF02591962>
- [38] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *International Journal of Robotics Research*, vol. 30, no. 2, pp. 236–258, 2011.
- [39] M. Vukobratovic, "Zero-moment pointâproper interpretation and new applications," 2001. [Online]. Available: http://www.robot-learning.de/pmwiki/uploads/Research/LocomotionSeminar/vukobratovic_humanoids2001.pdf
- [40] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. Center of pressure-zero moment point," *IEEE Transactions on Systems, Man, and*

- Cybernetics - Part A: Systems and Humans*, vol. 34, no. 5, pp. 630–637, 2004.
- [41] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, “Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point,” *Robotics and Automation (ICRA), 2003*, pp. 1620–1626, 2003.
- [42] A. W. Winkler, F. Farshidian, D. Pardo, M. Neunert, and J. Buchli, “Fast Trajectory Optimization for Legged Robots using Vertex-based ZMP Constraints,” pp. 1–8, 2017. [Online]. Available: <http://arxiv.org/abs/1705.10313>
- [43] G. Guennebaud, B. Jacob *et al.*, “Eigen v3,” <http://eigen.tuxfamily.org>, 2010.
- [44] L. D. Gasper, “Quadprog++,” <http://quadprog.sourceforge.net>.
- [45] M. L. Felis, “Rbd: an efficient rigid-body dynamics library using recursive algorithms,” *Autonomous Robots*, pp. 1–17, 2016. [Online]. Available: <http://dx.doi.org/10.1007/s10514-016-9574-0>
- [46] C. Gehring *et al.*, “Kindr-kinematics and dynamics for robotics,” Nov 2018.
- [47] O. S. R. Foundation, “Gazebo,” <http://gazebo.org>, 2014.
- [48] S. Russell, “Ode - open dynamics engine,” <http://www.ode.org>, 2001.
- [49] C. Dario Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, “Perception-less terrain adaptation through whole body control and hierarchical optimization,” *IEEE-RAS International Conference on Humanoid Robots*, pp. 558–564, 2016.
- [50] M. Bloesch, M. Burri, H. Sommer, R. Siegwart, and M. Hutter, “The Two-State Implicit Filter - Recursive Estimation for Mobile Robots,” *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 1–1, 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/8118116/>
- [51] P. Fankhauser, M. Bjelonic, D. Bellicoso, T. Miki, and M. Hutter, “Robust Rough-Terrain Locomotion with a Quadrupedal Robot,” *Icra*, no. May, p. 8, 2018.
- [52] H. Kolvenbach, D. Bellicoso, F. Jenelten, L. Wellhausen, and M. Hutter. ESA Conference Bureau, 2018-06-04.
- [53] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, “Learning agile and dynamic motor skills for legged robots,” *Science Robotics*, vol. 4, no. 26, 2019. [Online]. Available: <https://robotics.sciencemag.org/content/4/26/eaau5872>
- [54] DARPA, “Subterrain challenge,” <https://www.subtchallenge.com>, 2019.