Rapid Stability Margin Estimation for Contact-Rich Locomotion

Romeo Orsolino, Siddhant Gangapurwala, Oliwier Melon,
Mathieu Geisert, Ioannis Havoutis and Maurice Fallon

***Abstract*— The efficient evaluation the dynamic stability of legged robots on non-coplanar terrains is important when developing motion planning and control policies. The inference time of this measure has a strong influence on how fast a robot can react to unexpected events, plan its future footsteps or its body trajectory. Existing approaches suitable for real-time decision making are either limited to flat ground or to quasi-static locomotion. Furthermore, joint-space feasibility constraints are usually not considered in receding-horizon planning as their high dimensionality prohibits this.**

In this paper we propose the usage of a stability criterion for dynamic locomotion on rough terrain based on the Feasible Region (FR) and the Instantaneous Capture Point (ICP) and we leverage a Neural Network (NN) to quickly estimate it.

We show that our network achieves satisfactory accuracy with respect to its analytical counterpart with a speed up of three orders-of-magnitude. It also enables the evaluation of the stability margin's gradient. We demonstrate this *learned stability margin* in two diverse applications - Reinforcement Learning (RL) and nonlinear Trajectory Optimization (TO) for legged robots. We demonstrate on a full-sized quadruped robot that the network enables the computation of physically-realizable Center of Mass (CoM) trajectories and foothold locations satisfying friction constraints and joint-torque limits in a receding-horizon fashion and on non-coplanar terrains.

I. Introduction

Dynamic locomotion of legged robots on rough terrains is hard to achieve because of the high dimensionality of multi-limbed floating-base systems and the need for fast decisions about the footsteps placement and base trajectory. Typical model-based approaches consist of optimal control and TO formulations which either focus on quasi-static locomotion or use simplified dynamics' models to reduce the dimensionality of the system. These simplified models capture the main dynamics of the robotic platform but fail to grasp important feasibility constraints of legged robots such as the kinematic (leg workspace and joint-velocity) and dynamic (*e.g.,* friction cones and joint-torque) constraints. Instead of being used in an anticipative capacity, these limits are therefore usually only verified at the control level and can saturate in case of violation. However, these physical and hardware limits become significant when the robot negotiates challenging environments and gets closer to its performance envelope. When complex configurations are required, joint-level feasibility constraints cannot be neglected at the planning level [1].

This work was supported by the the UKRI/EPSRC RAIN [EP/R026084/1] and ORCA [EP/R026173/1] Hubs and the EU H2020 Projects MEMMO and THING, the EPSRC grant Robust Legged Locomotion [EP/S002383/1] and a Royal Society University Research Fellowship (Fallon). This work was conducted as part of ANYmal Research, a community to advance legged robotics. This work was part of the Human-Machine Collaboration Programme, supported by a gift from Amazon Web Services. The authors are with Dynamic Robots Systems (DRS) group, Oxford Robotics Institute, University of Oxford, UK. email: rorsolino, siddhant, omelon, mathieu, ioannis, mfallon@robots.ox.ac.uk

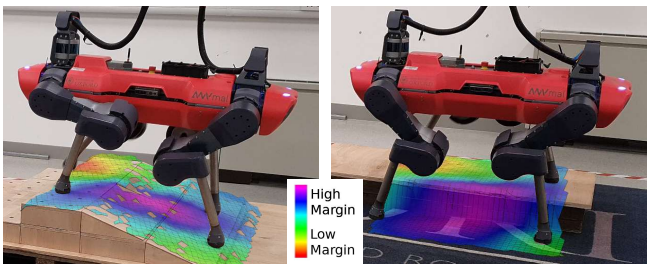

Fig. 1. The colormaps represent the value of the learned stability margin for different horizontal positions of the robot's base. High margin values (purple) correspond to $0.07\,\mathrm{m}$ whereas the low values (red) correspond to $-0.17\,\mathrm{m}$.

One approach to tackle the computational complexity of floating base systems is to project the joint-space constraints into lower-dimensional spaces, typically at the centroidal level (2-D or 6-D). Such projections are low-dimensional and preserve the descriptiveness of the original space but they are often expensive to compute thus defeating their purpose.

Along a different track of research, data-driven methods attempt to move the computational burden offline, thus interpreting the online decision process as a mere input-output relationship between state and the learned behavior. In this framework, RL methods have recently gained interest due to their reactiveness which allows them to retain stability on non-flat terrains [2], [3], [4]. Unfortunately, these approaches require a significant amount of offline data generation and training and are sensitive to reward function tuning.

A. Contributions

In this paper, we demonstrate how machine learning can allow fast evaluation of feasibility constraints (contact friction constraints and joint-torque limits) projected onto a 2-D space. The obtained network is a mathematical tool suitable for the *online* generation of dynamic locomotion in presence of non-coplanar contacts in different control settings. Our contributions can be summarized as follows:

- We leverage data-driven methods to evaluate dynamic stability of legged robots in the presence of non-coplanar contacts. For a small loss of accuracy, we gain a three-order-of-magnitude speed-up in the evaluation of the *stability margin* and the possibility to compute its gradient, which would otherwise be only possible by finite-difference of the analytical computation.

- we showcase applications of the mentioned tool in both TO and RL. We validate our approach in a physics-based simulator over multiple gaits (walk and trot) and on the hardware ANYmal C quadruped robot [5] in presence of non-coplanar scenarios.

II. Related Works

The fast assessment of the stability of a legged robot is a core issue in motion planning and control, especially for multi-contact scenarios [6], [7]. The fast binary assessment of static stability can be performed by means of Linear Programming (LP) [7] or Incremental Projection (IP) algorithms [6]. Optimal control and TO approaches, however, require a quantitative–and *non-binary*–measure of robustness [8].

Only a few works have focused on dynamic stability on complex terrains, mostly through the generalization of the *capturability analysis* [9] to height variations [10] or to the 3-D space of the CoM position [11], [12]. Other approaches have instead focused on the generalization of the Zero Moment Point (ZMP) onto the 6-D centroidal space [13], [14]. Some of them have also included the possibility to synthesize body trajectories that avoid joint-torque limits [15]. All of them, however, assume the contact locations to be predetermined and they are therefore not suitable for footstep planning on rough terrains.

2-D projections of such sets are possible [16], which enables sampling-based footstep planning on rough terrains [17]. Footstep planning has also been performed through A* based algorithms [18], mixed-integer convex programming [19] or LP [20]. Such methods, however, they all consider predefined base trajectories. Simultaneous optimization of base trajectories and contact locations on non-coplanar terrains has been achieved using mixed-integer programming [21] and nonlinear TO [22] but their computation time does not allow these strategies to be deployed in a receding-horizon fashion.

Approaches based on machine learning, on the other hand, have demonstrated their ability to generate dynamic locomotion behaviors on both flat grounds [23] and rough terrains [2]. These approaches, however, require long offline training phases and accurate reward function tuning. Tsounis *et al.* [3] leverage a model-based feasibility method [24] to guarantee the feasibility of their policy and to discard physically-unrealizable trajectories. Our RL application presented in this paper, presents similarities with this work in the sense that we also leverage a dense reward function to improve the sampling efficiency of our training phase; the reward function is in our case, however, an estimate of the analytical feasibility measure learned by means of a NN. Learning of the robot's centroidal dynamics has already been proposed to speed up motion planners on rough terrain [25], [26]. However, to the best of our knowledge, it is the first time that a learned dynamic feasibility constraint is demonstrated on the hardware for multiple gaits and multiple terrain.

On a similar line to our approach, Carpentier *et al.* [27] proposed to learn feasibility constraints in order to overcome the computation burden and enable online planning on complex terrain. Differently from our work, however, they focused on kinematic limits and learned a probability density function of the CoM positions whereas we focus on friction constraints and joint-torque limits. Besides that, we demonstrate the applicability of our method for simultaneous optimization of base trajectories and contact locations in two different settings: nonlinear TO and RL.

III. Background

For completeness, we summarize in the following the steps required to compute the feasible region [17] and the ICP. Based on these quantities we then define *analytical* stability margin, a mathematical tool which we employ in this paper to quantify the dynamic stability of legged robots during motion over rough terrain. In Sec. IV we go on to explain how we use a data-driven method to offset the computational complexity of this analytical approach.

A. Feasible Region

Bretl *et al.* [6] demonstrated how the support region differs from the convex hull of contacts points when the robot establishes non-coplanar contacts with the environment. The authors demonstrated how the IP algorithm can be employed to compute the support region and to efficiently test the stability of the robot [6], [28]. Subsequently, this method was extended to account for joint-torque limits [17] and to dynamic conditions such as underactuated support configurations (single and double point contacts) and external wrenches [1].

A summary of this improved strategy is given in Alg. 1. The algorithm iteratively solves a LP whose optimization variables are the CoM projection $\mathbf{c}_{xy} \in \mathbb{R}^2$, the contact forces $\mathbf{f}_i \in \mathbb{R}^3$ and the contact torques $\mathbf{w}_i \in \mathbb{R}^2$. The LP seeks the furthest point $\mathbf{c}^*_{xy}$ along the $\mathbf{a}_j \in \mathbb{R}^2$ direction on the (x, y) plane that satisfies dynamic equilibrium (III.a) while the contact forces of each end-effector i in contact with the

Algorithm 1 IP algorithm.

Input: $\boldsymbol{\lambda} \in \mathbb{R}^{47}$
Result: feasible region $\mathcal{Y}_{fr}$
Initialization: $\mathcal{Y}_{outer}$, $\mathcal{Y}_{inner}$, get $\mathbf{q}$ given $\boldsymbol{\lambda}$ through IK.
while $area(\mathcal{Y}_{outer}) - area(\mathcal{Y}_{inner}) > \epsilon$ **do**
I) compute the edges $\mathbf{e}_j$ of $\mathcal{Y}_{inner}$
II) pick $\mathbf{a}_j$ cutting off the largest fraction of $\mathcal{Y}_{outer}$
III) solve the LP:

$$\mathbf{c}^*_{xy} = \underset{\mathbf{c}_{xy}, \mathbf{f}_i, \mathbf{w}_i}{\operatorname{argmax}} \quad \mathbf{a}_j^T \mathbf{c}_{xy}$$

such that :

$$\text{dynamic equilibrium} \qquad \text{(III.a)}$$
$$\mathbf{f}_i \in \mathcal{C}_i, \quad \forall i = \{1, 2, \dots N_c\} \qquad \text{(III.b)}$$
$$\mathbf{f}_i \in \mathcal{F}_i, \quad \forall i = \{1, 2, \dots N_c\} \qquad \text{(III.c)}$$
$$\mathbf{w}_i \in \mathcal{B}_i, \quad \forall i = \{1, 2, \dots N_c\} \qquad \text{(III.d)}$$

IV) update the outer approximation $\mathcal{Y}_{outer}$
V) update the inner approximation $\mathcal{Y}_{inner}$
end while

environment lie within their friction cone $\mathcal{C}_i$ (III.b):

$$\mathcal{C}_i = \left\{ \mathbf{f}_i \in \mathbb{R}^3 \quad | \quad ||(\mathbb{1} - \mathbf{n}_i \mathbf{n}_i^T)\mathbf{f}_i||_2) \leq \mu \mathbf{n}_i^T \mathbf{f}_i \right\} \tag{1}$$

and force polytope $\mathcal{F}_i$ (III.c):

$$\begin{aligned} \mathcal{F}_i = \Big\{ \mathbf{f}_i \in \mathbb{R}^3 | \quad \exists \boldsymbol{\tau}_i \in \mathbb{R}^{n_a} s.t. \quad \mathbf{M}_{bi}^T \dot{\boldsymbol{\nu}}_b + \mathbf{M}_i \ddot{\mathbf{q}}_i + \\ \mathbf{c}(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{g}(\mathbf{q}_i) = \boldsymbol{\tau}_i + \mathbf{J}(\mathbf{q}_i)^T \mathbf{f}_i, \quad \underline{\boldsymbol{\tau}}_i \leq \boldsymbol{\tau}_i \leq \overline{\boldsymbol{\tau}}_i \Big\} \end{aligned} \tag{2}$$

and while the contact torques $\mathbf{w}_i$ satisfy the box constraint $\mathcal{B}_i$ (III.d):

$$\mathcal{B}_i = \left\{ \mathbf{w}_i \in \mathbb{R}^2 \quad | \quad \underline{\mathbf{w}}_i \leq \mathbf{w}_i \leq \overline{\mathbf{w}}_i \right\} \tag{3}$$

$\mu \in \mathbb{R}$ represents the friction coefficient between the robot's foot and the ground (assumed here to be the same for all contacts). The vectors $\mathbf{q}_i, \dot{\mathbf{q}}_i$ and $\ddot{\mathbf{q}}_i \in \mathbb{R}^{n_a}$ represent the joint position, velocity and acceleration of the i^{th} limb of the articulated robot. $\boldsymbol{\nu}_b = [\dot{\mathbf{c}}^T, \boldsymbol{\omega}^T]^T \in \mathbb{R}^6$ represents the generalized base velocity. The matrices $\mathbf{M}_{bi}, \mathbf{M}_i$ and $\mathbf{J}(\mathbf{q}_i)$ represent, respectively, the inertia of the base, the inertia of the legs, and the Jacobian. The vectors $\underline{\boldsymbol{\tau}}_i, \overline{\boldsymbol{\tau}}_i \in \mathbb{R}^{n_a}$ represent the minimum and maximum joint-torque limits of the n_a actuators of one individual leg i. $\mathbf{w}_i = [w_x, w_y]^T \in \mathbb{R}^2$ is an infinitesimal contact torque (decoupled from the contact forces) having maximum value of $\overline{\mathbf{w}}_i = [w_M, w_M]^T \in \mathbb{R}^2$ and minimum value of $\underline{\mathbf{w}}_i = -\overline{\mathbf{w}}_i$ such that $w_M \in \mathbb{R} \to 0$. The torques $\mathbf{w}_i$ enable the computation of the feasible region also in presence of single or double point contacts [1].

The set of all variables required to quantify the robot's instantaneous dynamic balance are the following:

$$\begin{aligned} \boldsymbol{\lambda} = (\underbrace{\mathbf{e}_g}_{\mathcal{B}}, \underbrace{\dot{\mathbf{c}}, \ddot{\mathbf{c}}, \dot{\boldsymbol{\omega}}, \mathbf{f}_{ext}, \boldsymbol{\tau}_{ext}, \mathbf{p}_1, \dots \mathbf{p}_{N_c}}_{\mathcal{H}}, \\ \mu, c_1, \dots c_{N_c}, \underbrace{\mathbf{n}_1, \dots \mathbf{n}_{N_c}}_{\mathcal{H}}) \in \mathbb{R}^{47} \end{aligned} \tag{4}$$

$\mathbf{e}_g \in \mathbb{R}^3$ is a unit vector representing the gravity axis in the base frame $\mathcal{B}$, *i.e.,* the third row of the robot's base orientation matrix $\mathbf{R} \in SO(3)$. $\dot{\mathbf{c}}, \ddot{\mathbf{c}} \in \mathbb{R}^3$ are the linear velocity and acceleration of the robot's CoM whereas $\dot{\boldsymbol{\omega}} \in \mathbb{R}^3$ is the angular acceleration of the robot's base. $\mathbf{f}_{ext}, \boldsymbol{\tau}_{ext} \in \mathbb{R}^3$ compose the wrench of an external disturbance applied on the base of the robot (if any). $\mathbf{p}_i \in \mathbb{R}^3$, $\mathbf{n}_i \in \mathbb{R}^3$ and $c_i \in \{0, 1\}$ for $i = \{1, \dots N_c\}$ represent the feet positions, the contact normals and the contact state, respectively. Each of these quantities provided in the *horizontal* frame[1] as indicated in (4). It is important to notice that an implicit information about the robot's CoM position $\mathbf{c}$ is encoded in $\boldsymbol{\lambda}$ thanks to the definition of the feet positions $\mathbf{p}_i$ with respect to $\mathcal{H}$. The joint-space variables required in (2), but missing from (4), can be determined from $\boldsymbol{\lambda}$ through Inverse Kinematics (IK). For articulated robots with non-redundant limbs, such as ANYmal C, the IK problem can be efficiently solved analytically.

[1]The horizontal frame has been also called with multiple other names in the literature such as *control*, *local* or *yaw* frame [29]. It differs from the *global* frame because of its different location and yaw angle. It differs from the robot's *base* frame for its different roll and pitch angles.

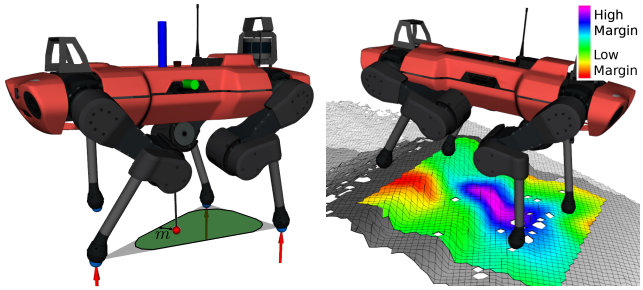

Fig. 2. Left: we define the stability margin as the perpendicular distance from the contour of the feasible region (green region) and the ICP (red sphere). Right: colormap representing the values of the learned stability margin where purple is the highest and red is the lowest.

At each iteration of Alg. 1 a new vertex $\mathbf{c}_{xy}^*$ is added to the inner approximation $\mathcal{Y}_{inner}$ and a new edge orthogonal to $\mathbf{a}_j$ and passing through $\mathbf{c}_{xy}^*$ is added to $\mathcal{Y}_{outer}$ [6].

Once $\mathcal{Y}_{inner}$ and $\mathcal{Y}_{outer}$ have converged to the desired tolerance ϵ, we obtain the feasible region $\mathcal{Y}_{fr}$ as the *halfspace-description* of $\mathcal{Y}_{inner}$, consisting of a number $N_e \in \mathbb{N}$ of edges $\mathbf{e}_j \in \mathbb{R}^2$ with $j = 1, \dots, N_e$.

B. Capture Point

The ICP $\boldsymbol{\xi} \in \mathbb{R}^2$ has been defined by Pratt *et al.* [9] as the point on the flat ground where a robot, given its instantaneous linear momentum, should place its foot to come to a complete stop:

$$\boldsymbol{\xi} = \mathbf{c}^{xy} + \frac{\dot{\mathbf{c}}^{xy}}{\omega} \tag{5}$$

where $\omega = \sqrt{g/c_z}$ is the natural frequency of the Linear Inverted Pendulum (LIP) model while $\mathbf{c}^{xy}$ and $\dot{\mathbf{c}}^{xy}$ are the projection on the horizontal (x, y) plane of the CoM position and velocity. The ICP $\boldsymbol{\xi}$ can also be considered as the point where the CoM projection is converging to [9]. We employ it as a ground reference point to determine the instantaneous stability of our system: the robot is stable if: $\boldsymbol{\xi} \in \mathcal{Y}_{fr}$.

C. Analytic Stability Margin

The feasible region is a convex set by construction [17]; we can compute the normals $\mathbf{n}_{e,j} \in \mathbb{R}^2$ to its edges and the offset terms $a_j \in \mathbb{R}$ such that $\mathbf{n}_{e,j} \cdot \mathbf{p} - a_j = 0$ if $\mathbf{p} \in \mathbb{R}^2$ is a point that lies along $\mathbf{e}_j$. The normals can be stacked into a matrix $\mathbf{N}$:

$$\mathbf{N} = \begin{bmatrix} \mathbf{n}_{e,1}^T & a_1 \\ \vdots & \vdots \\ \mathbf{n}_{e,N_e}^T & a_{N_e} \end{bmatrix} \in \mathbb{R}^{N_e \times 3} \tag{6}$$

so that the signed distances $\mathbf{d} \in \mathbb{R}^{N_e}$ between a 2-D point $\mathbf{p}$ and all the edges of $\mathcal{Y}_{fr}$ can be computed as: $\mathbf{d} = \mathbf{N} \cdot [\mathbf{p}^T \quad 1]^T$. If all the normals $\mathbf{n}_i$ are pointing inwards then all the elements of $\mathbf{d}$ will be positive if $\mathbf{p}$ lies within $\mathcal{Y}_{fr}$. We can then define the *analytic* stability margin m_a (see Fig. III-B) as the minimum distance between the ICP $\boldsymbol{\xi}$ and the edges of the feasible region:

$$m_a = \min(\mathbf{N} \cdot [\boldsymbol{\xi}^T \quad 1]^T) \tag{7}$$

This will result into a positive value if $\boldsymbol{\xi}$ lies inside the feasible region and in a null or negative value otherwise. In Sec. IV we will explain how we used a NN to estimate this quantity from data.

IV. Learned stability margin

The algorithm presented in the previous section depends on the centroidal and contacts state of the robot (from which also the required joint-space states needed to determine the force polytopes can be inferred).

The vector $\boldsymbol{\lambda} \in \mathbb{R}^{47}$ represents the input state of a Multi-Layer Perceptron (MLP) that we employed to estimate the output of the IP algorithm that we described in Section III-C. The learned stability margin m_l can thus be obtained as:

$$m_l = MLP(\boldsymbol{\lambda}) \tag{8}$$

A. Learning

We followed the following procedure to train the network:

1) Network Architecture: The dense feed-forward network used as the function approximator of the stability margin comprises the 47-dimensional input $\boldsymbol{\lambda}$, 1-dimensional output m_l, and 3 hidden layers (256 nodes in the first layer and 128 nodes in the other two layers). We use the softsign activation function to introduce non-linearity in the network.

2) Generating Training Data: We use the parameters represented in Table I as input to the network. In order to generate the training set we sample these parameters from the distributions shown in Table I and use the IP algorithm to compute the analytical stability margin m_a for each set of the parameters. Note that whereas we do account for angular accelerations of the robot's base, the base angular velocity $\boldsymbol{\omega}$ is the only dynamic parameter that is not accounted for in our strategy, it does not belong to input vector $\boldsymbol{\lambda}$ and therefore no random data generation is required for this variable.

We noted that the usage of a uniform distribution for the feet locations and for the base velocity and acceleration (x, y components) was particularly useful to make sure that the accuracy of the trained network would not quickly deteriorate whenever the robot deviated significantly from its nominal configuration (in which all the joint-positions are equidistant from their kinematic end-stops) or from static conditions.

3) Supervised Learning: We then trained the network by generating a dataset of $50 \cdot 10^6$ samples using a supervised learning approach employing the Adam optimization strategy [30] for backpropagation.

A graphic representation of the estimates returned by the network is given in Fig. 2 (right). In addition to the substantial speed up in the inference time, the so obtained MLP also enables partial derivative computation of the output (the stability margin) with respect to the input state. As we will in next Section, this leads to the possibility to directly employ the MLP in nonlinear optimization problems whose variables depend on the elements of $\boldsymbol{\lambda}$.

B. Nonlinear Trajectory Optimization

To demonstrate the applicability of the learned stability margin m_l in a motion planning setting, we have formulated a nonlinear TO problem which leverages the proposed MLP to enable the concurrent optimization of the robot's base trajectory (pose, linear and angular velocities) and contact locations on arbitrary terrains in a receding-horizon fashion:

$$\underset{\mathbf{x}_b(t_k), \boldsymbol{\nu}_b(t_k), \mathbf{p}_i(t_j)}{\operatorname{argmin}} \quad \underbrace{w \left|\left| \frac{\partial h_t}{\partial \mathbf{p}_i^{xy}(t_j)} \right|\right|_2^2}_{\mathcal{C}}$$

such that :

$$\begin{aligned}
MLP(\boldsymbol{\lambda}(t_k)) \geq & m_{min} && \text{(minimum stability margin)} \\
\mathbf{p}_i(t_j) \leq & \mathcal{R}(\mathbf{x}_b) && \text{(foot range of motion)} \\
p_i^z(t_j) = & h_t(\mathbf{p}_i^{xy}) && \text{(terrain height)} \\
\mathbf{x}_b(t_0) = & \mathbf{x}_{t_0,b}, \quad \boldsymbol{\nu}_b(t_0) = \boldsymbol{\nu}_{t_0,b} && \text{(initial state)} \\
\mathbf{x}_b(t_N) = & \mathbf{x}_{T,b}, \quad \boldsymbol{\nu}_b(t_N) = \boldsymbol{\nu}_{T,b} && \text{(target state)}
\end{aligned} \tag{9}$$

where $\mathbf{x}_b = [\mathbf{c}, \mathbf{R}] \in SE(3)$ is the pose of the robot's base link and $\mathbf{R} \in SO(3)$ is the base orientation. The subscript i represents the end-effector indices $\{1, 2, \dots N_c\}$ (where N_c is the number of contacts) whereas the timestamps t_k (with $k = 1, \dots, N$) represent the time instants of the base discretization nodes, equally spaced along the optimization horizon. For the feet variables, rather than using a uniformly spaced discretization like the base variables, an event-based parametrization has been performed, only setting a foot node at each touch down instant t_j and keeping that variable fixed throughout the stance phase duration ($j = 1, \dots N_{s,i}$ where $N_{s,i}$ is the predefined number of steps of the i^{th} foot over the horizon T). The problem has been formulated using third-order polynomials to ensure continuity between the nodes. The feet sequence here is predefined and so is the duration of each stance and swing phase. The vertical coordinate of the

TABLE I

Network input parameters and the sampling distributions used for generating training data.

Parameter	Distribution
Feet in contact	$X \sim U(0, 3)$
Base roll and pitch angles	$\alpha, \beta \sim U(-\pi/15, \pi/15)$
COM linear velocity	$\dot{x}, \dot{y} \sim U(-1.25, 1.25)$ $\dot{z} \sim \mathcal{N}(0, 0.2)$
COM linear acceleration	$\ddot{x}, \ddot{y} \sim U(-2.5, 2.5)$ $\ddot{z} \sim U(-1.25, 1.25)$
Base angular velocity	–
Base angular acceleration	$\dot{\omega}_x, \dot{\omega}_y \sim U(-1.0, 1.0)$ $\dot{\omega}_z \sim \mathcal{N}(0, 0.5)$
Friction coefficient	$\mu \sim U(0.2, 0.9)$
$i-th$ foot deviation from nominal position	$p_i^x \sim U(-0.35, 0.35)$ $p_i^y, p_i^z \sim U(-0.2, 0.2)$
Contact normal rotations (about gravity axis)	$\alpha \sim \mathcal{N}(0, \pi/8)$ $\beta \sim \mathcal{N}(0, \pi/8)$
External force	$F_x, F_y, F_z \sim \mathcal{N}(0, 50)$
External torque	$\tau_x, \tau_y, \tau_z \sim \mathcal{N}(0, 25)$

footholds p_i^z is set equals to h_t, as enforced by the *terrain height* constraint. The cost $\mathcal{C}$ finds a foothold on the 2.5-D map that minimizes the terrain slope which enables the robot to avoid inclined surfaces and edges as it navigates over rough terrain.

The *minimum stability margin* constraint is used to enforce dynamic feasibility of the trajectory all along the optimization horizon. The value of m_{min} is gait dependent: in the trot gait, for example, the feasible region is a 1-D segment and the maximum achievable value will thus be zero (when the capture point $\boldsymbol{\xi}$ belongs to the segment). A minimum margin value $m_{min} = -0.05$, for example, will guarantee that the capture point does not diverge during the optimization horizon and that, consequently, the robot does not fall. This constraint is computed using the learned stability margin m_l from the dynamic state of the robot (4) at each base's discretization nodes. As mentioned above, the derivative of m_l with respect to the optimization variables, which is necessary for the solution of our TO problem, can be obtained as a function of the of the vector $\partial m_l / \partial \boldsymbol{\lambda}$, by backpropagation of the MLP. Consider, the CoM position variable $\mathbf{c}$, expressed in (9) with respect to the world frame $\mathcal{W}$. Although $\mathbf{c}$ is not part of the MLP's input state $\boldsymbol{\lambda}$, the partial derivative $(\partial m / \partial \mathbf{c})$ can still be obtained as a function of the feet positions $\mathbf{p}$ expressed in the horizontal frame $\mathcal{H}$ (which are part of $\boldsymbol{\lambda}$):

$$\frac{\partial m}{\partial \mathbf{c}} = \sum_{i=1}^{4} \left(\frac{\partial m}{\partial \mathbf{p}_i} \underbrace{\frac{\partial \mathbf{p}_i}{\partial \mathbf{c}}}_{=-1} \right) = -\sum_{i=1}^{4} \frac{\partial m}{\partial \mathbf{p}_i} \tag{10}$$

Because, by definition, the horizontal frame is located in the CoM, a translation of $\mathbf{c}$ will result in a translation of $\mathbf{p}$ of equal amplitude in the opposite direction, which gives $\partial \mathbf{p} / \partial \mathbf{c} = -1$. Moreover, considering that we do not know a priori which foot affects the margin m for a given motion, we need therefore to sum up the contributions of each leg.

Note that the *minimum stability margin* is the sole term in the proposed TO problem that enforces dynamic consistency. Because the MLP has already learned offline the relationship between task space variables and feasible state of the robot, it is not necessary to include here the contact forces as optimization variables or the feasibility constraints already included in Alg. 1 such as friction cones and force polytopes.

C. Reinforcement Learning

The proposed learned stability margin can also be used to obtain a control policy trained using model-free approaches such as RL [4], [31].

Most of the existing RL approaches for the complex task of legged locomotion are sample inefficient and require significant amount of simulated runs. Therefore, it becomes necessary to implement efficient RL environment setups in order to minimize computation time of each simulation step thereby reducing the total training time. However, in order to obtain a RL policy which exhibits the desired locomotion behavior, significant platform-specific engineering is often necessary. This is usually done by tuning a dense reward function: utilizing the stability margin m_a as a metric of the robot's performance in the reward function enables the use of such a dense function. Additionally, learning the stability margin estimate m_l allows for the necessary reduction of the stability inference by a considerable amount. In Sec. V-C we show simulation results of how a RL policy, extensively described in [4], can benefit from the leverage of the learned stability margin.

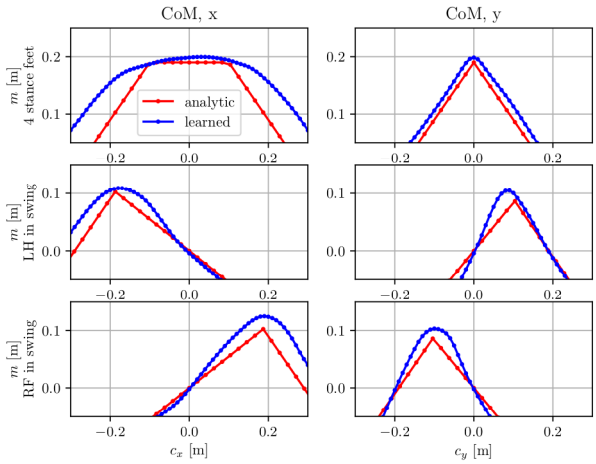

Fig. 3. Comparison of the learned stability margin m_l (blue) and its analytical counterpart m_a (red). Variation of the stability margin w.r.t.x (left) and y (right) coordinates of the CoM for a quadruple (top) and triple (middle and lower) stance configuration. The learned margin tends to divert from the analytical margin far away from the training distribution (see Tab. I) and near the discontinuities of the analytical margin's gradient (see Fig. 4), which is a necessary trade-off to achieve a smooth m_l function.

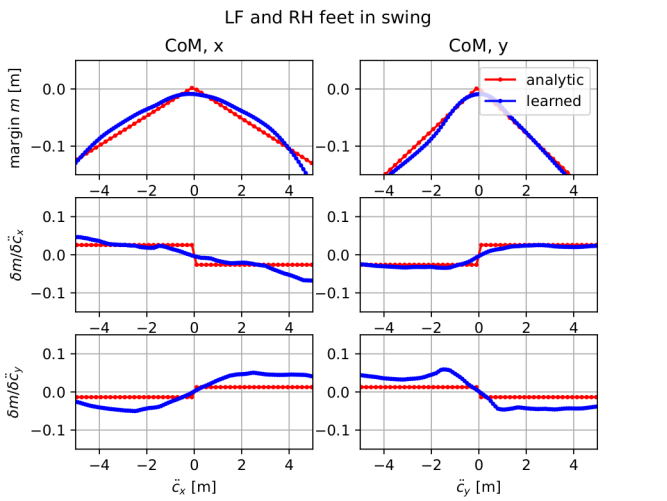

Fig. 4. Comparison of the stability margin (top) and its partial derivatives (middle and bottom): the analytical and learned partial derivatives were obtained using finite differences and backpropagation, respectively. Variation of the stability margin (top) and of its partial derivatives $\partial m / \partial \ddot{c}_x$ (middle) and $\partial m / \partial \ddot{c}_y$ (bottom) with respect to x (left) and y (right) coordinates of the CoM linear acceleration for a double stance configuration.

V. Experimental Results

The learned stability margin m_l can be evaluated in about 10 µs thus offering a significant computational speed-up

Fig. 5. Snapshots of a Gazebo simulation where the ANYmal C robot walks over obstacles of 6, 12 and 18 cm while replanning its base pose, its linear and angular velocity and contact locations at 1.5 Hz adapting to the latest height map using the TO formulation presented in Sec. IV-B.

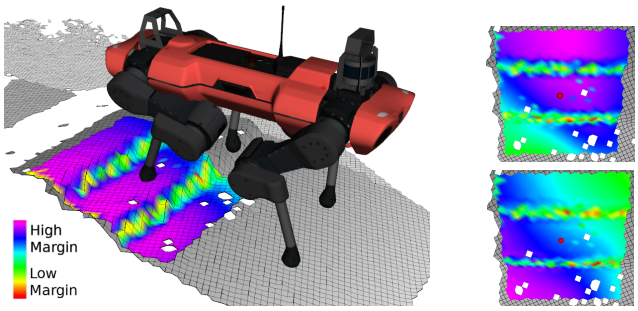

Fig. 6. Values of the learned stability margin when we vary the position of the left-front (LF) leg at different time of the walking gait cycle. Left: at full stance; top right: during swing of the right-front (RF) leg; bottom right: during swing of the left-hind (LH) leg. The prevalent purple color indicates that the motion planner has selected a good foothold as the stability margin value remains above zero throughout the whole gait cycle.

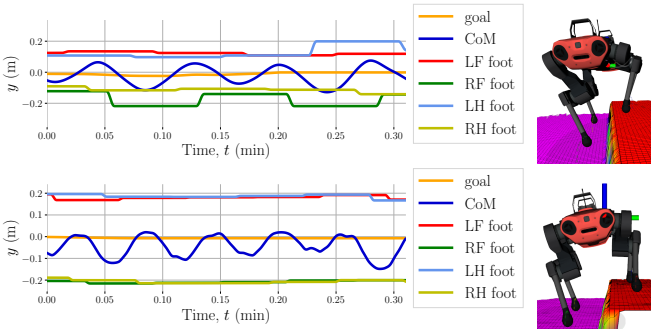

Fig. 7. Comparison of receding-horizon motion planning on a sagittal step of 18 cm using the SRB model (top) and the proposed feasible region (bottom). Using the feasible region results in wider stance configurations without needing of additional costs/reward functions and it pushes the CoM further away from the most retracted legs, in order to alleviate the load on those joints when close to their torque limits.

compared to the analytical counterpart. In the next Sections we analyse the performances of the network in terms of reliability and accuracy. We then report the experimental results obtained in simulation and on the hardware using the nonlinear TO formulation and the RL policy described above.

A. Reliability and Accuracy

We measured reliability as the amount of true positives and true negatives that were estimated by the network when assessing the stability of the robot's state as a binary problem, considering a state to be stable when $m_l \geq 0$. We evaluated the models predictability based on the range of samples within two times the standard deviation of the dataset generated using the parameters sampled from the distributions given in Tab. I. Over a batch of $2.5 \cdot 10^4$ samples we obtained a true positive rate of 0.970 and a true negative rate of 0.984.

As regards the accuracy of the network, Fig. 3 shows a comparison between the analytical stability margin m_a (red) and the learned stability margin m_l (blue) in different stance configurations: four stance (top), triple stance with left-hind (LH) foot in swing (middle) and triple stance with right-front (RF) foot in swing (bottom). Figure 4 shows the evolution of the analytic (red) and learned (blue) stability margins (top) and partial derivatives(middle and bottom) over a wide range of CoM linear accelerations $\ddot{c}_x$ (left) and $\ddot{c}_y$ (right). The IP algorithm does not provide the partial derivatives of the analytical margin m_a, which we therefore computed by first-order central difference. This requires two calls to Alg. 1 resulting in a $20.0\,\mathrm{ms}$ computation time for each partial derivative; this leads to a $940.0\,\mathrm{ms}$ computation time for the vector of partial derivatives $\partial m_a / \partial \boldsymbol{\lambda}$ with respect to the full input state $\boldsymbol{\lambda} \in \mathbb{R}^{47}$, almost four orders of magnitude more than the time required to infer the learned counterpart $\partial m_l / \partial \boldsymbol{\lambda}$ (around $100\,\mu\mathrm{s}$).

The accuracy of the m_l curves shown in Fig. 4 is critical for its successful deployment in real-world applications. In particular, the derivatives with respect to the base acceleration are relevant for TO strategies. Although the accuracy error deteriorates as the state of the robot moves away from the training range (see Tab. I), we can see that the prediction error of m_l is always below $1.0\,\mathrm{cm}$ for CoM accelerations within the range of: $\ddot{c}_x \in [-2.0, +2.0]\ \mathrm{m/sec^2}$ and $\ddot{c}_y \in [-2.0, +2.0]\ \mathrm{m/sec^2}$.

Fig. 1 shows the value of the learned stability margin when varying the base position. As expected, the stability margin is larger near the centre of the support polygon but the largest margin does not necessarily correspond to this point since the margin used also takes into account the joint-torque constraints. Similarly, Fig. 6 shows the value of the learned margin when changing the position of the left front foot, at different time instants of the gait cycle. Since the used stability criterion takes into account the contact normals $\mathbf{n}_i$, the positions very close to the edges of steps are already naturally avoided without need to add any cost.

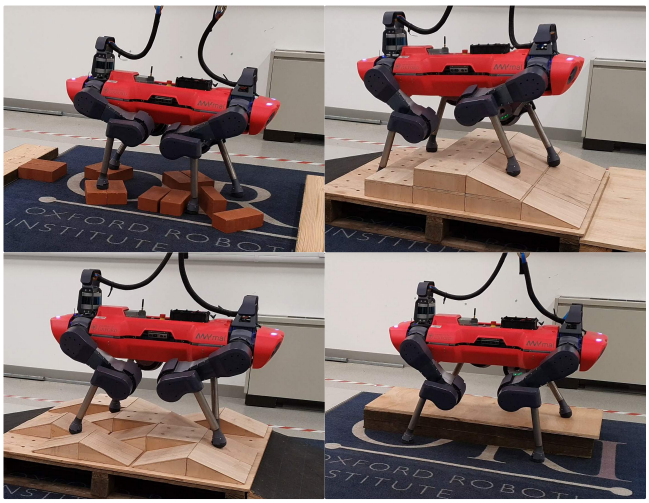

Fig. 8. Oxford's ANYmal C robot walks over bricks (top left), steps (top right), inclines (bottom left) and side steps (bottom right) using the receding-horizon nonlinear TO formulation proposed in Sec. IV-B using the learned stability margin as many feasibility criterion.

The most stable foot position change along the gait cycle, so the best contact position can be selected by computing the margin over the whole cycle. Next section shows this can be achieved through trajectory optimization.

B. Nonlinear Trajectory Optimization

To demonstrate the applicability of the proposed network TO, we tested the formulation described in Section IV-B using the ANYmal C robot in simulation (Gazebo) and experimentally. We used a time horizon T of $3.0\,\mathrm{s}$, a base trajectory discretization time of $\Delta T = 0.2\,\mathrm{s}$ and the gait parameters corresponding to a walk gait (with feet trajectory nodes discretized at touch down and lift-off). This resulted in an overall problem of 200 optimization variables and 500 constraints. Convergence took about 9 iterations using the Interior Point nonlinear solver IPOPT [32].

On a 4-core/8-thread Intel Xeon(R) CPU E3-1505M v6 @ $3.00\,\mathrm{GHz}$ computer this corresponded to about $70\,\mathrm{ms/iter}$. On the robot this fell to $40\,\mathrm{ms/iter}$. We ran the motion planner in a receding-horizon fashion at the frequency of $1.0\,\mathrm{Hz}$. As can be seen in Fig. 5 and 8, the motion planner enables the ANYmal C robot to adapt its body trajectory and footsteps location to the incoming terrain information from a 2.5-D height map with slopes and obstacles between 6 and $18\,\mathrm{cm}$.

The benefit of measuring the stability of the robot using the feasible region is particularly apparent in load-intensive tasks or in scenarios that require the robot to take on complex body configurations. In Fig. 7, for example, the ANYmal C is asked to walk along a side step, causing its left feet to be on a surface $18\,\mathrm{cm}$ above its right feet. In this case any simplified model, such as the Single Rigid Body (SRB) used to generate the motion shown in Fig. 7 (top), would require the CoM to be in the middle between the left and right feet, exactly as if they were all on the same flat surface. The feasible region (as in Alg. 1), instead, considers the joint-torque limitations of the robot and, for this reason, brings the CoM closer to the most extended legs which are in a more favourable configuration and therefore capable of carrying more body-weight (Fig. 7, bottom). Compared to the motion planning formulation based on the SRB model [33], the feasible region also leads the robot to walk with a wider stance without the need of any additional cost.

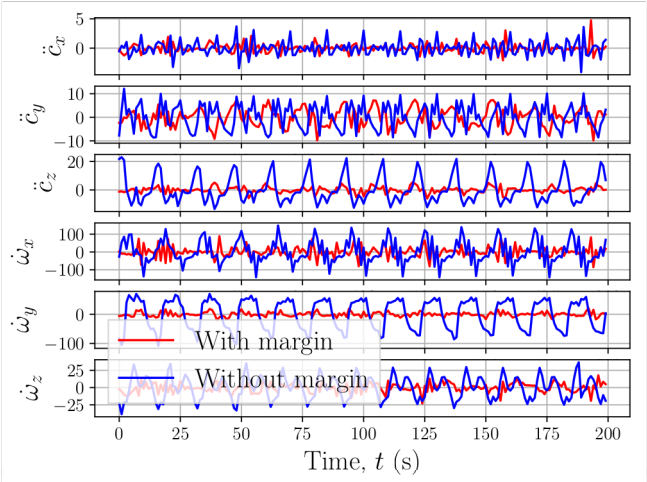

Fig. 9. Base acceleration trajectories resulting from two different RL policies for quadrupedal running trained without (blue) and with (red) a reward function maximising the learned stability margin m_l. This maximisation term results in smaller base accelerations.

C. Reinforcement Learning

The proposed learned stability margin m_l was also used in [4] to train a terrain-aware footstep planning policy, a domain adaptive tracking policy, and also an emergency recovery policy. This work proposed the use of m_l as a dense reward function in order to assess the policies performance for dynamic locomotion over uneven terrain. The work further compared the influence of the stability margin on the robot's stance; a higher coefficient used with the stability margin reward term resulted in a behaviour with wider stance, whereas a smaller coefficient resulted in a preference for narrower stance during stable phases of motion while switching to wider stance during unstable phases.

In our experiments, we observed that introducing the stability metric in the reward function enabled us to obtain smoother motion trajectories. We trained two RL policies which mapped the robot state information to desired joint state of the ANYmal B robot for the task of following a reference base velocity command. We trained a RL policy using a simple reward function given by, $L(8.0 \times (0.75 - \dot{x}))$ where $L(x)$ is the logistic kernel given by $L(x) = (e^{-x} + 2 + e^{x})^{-1}$ and $\dot{x}$ is the heading velocity of the base represented in the base frame. We compared the learned behaviour of this policy with that of another RL policy obtained using a reward function comprising the previously introduced logistic kernel term, further augmented using the stability margin such that the RL policy attempts to track the desired base reference velocity while also maximizing the stability margin. As represented in the plot in Fig. 9, it is evident that the policy obtained by introducing the stability margin in the reward term results in smoother base motions

and smaller base accelerations (red) compared to the case without margin maximization (blue). Furthermore, similarly to what observed for the TO application (see Fig. 7), we observed a preference for wider stance in the case of the policy augmented with the stability margin. This is a direct result of the stance allowing for a larger support region thereby maximising the stability margin.

VI. Conclusion and future works

In this paper, we have proposed the signed distance between the ICP and the closest edge of the feasible region [17], [1] as a measure of dynamic stability for legged robots in contact-rich scenarios and we have explained how to leverage a NN to overcome the computation time limitations of this measure. We have then reported the simulation and hardware results on the full-sized ANYmal C robot of two distinct applications in the fields of nonlinear TO and RL. In the TO setting, in particular, we have demonstrated how the learned stability margin can lead not only to the *online* concurrent planning of base trajectories and contact locations but also to the optimization of the body posture in order to avoid unnecessary violation of the robot's joint-torque limits.

We hope to advance this approach by including the learning of the joint-position limits [1] in such a way to embed into the TO and RL settings an accurate description of the body workspace.

References

[1] A. Abdalla, M. Focchi, R. Orsolino, and C. Semini, "An Efficient Paradigm for Feasibility Guarantees in Legged Locomotion," 2019. [Online]. Available: http://arxiv.org/abs/2011.07967

[2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, 2020. [Online]. Available: https://robotics.sciencemag.org/content/5/47/eabc5986

[3] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter, "Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3699–3706, 2020.

[4] S. Gangapurwala, M. Geisert, , R. Orsolino, M. Fallon, and I. Havoutis, "RLOC: Terrain-Aware Legged Locomotion using Reinforcement Learning and Optimal Control," 2020. [Online]. Available: http://arxiv.org/abs/2012.03094

[5] [Online]. Available: https://www.anybotics.com/anymal-legged-robot/

[6] T. Bretl and S. Lall, "Testing Static Equilibrium for Legged Robots," *IEEE Transactions on Robotics (TRO)*, vol. 24, no. 4, pp. 794–807, 2008. [Online]. Available: http://dx.doi.org/10.1109/TRO.2008.2001360

[7] A. Del Prete, S. Tonneau, and N. Mansard, "Fast algorithms to test robust static equilibrium for legged robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[8] N. Giftsun, A. Del Prete, and F. Lamiraux, "Robustness to Inertial Parameter Errors for Legged Robots Balancing on Level Ground," in *International Conference on Informatics in Control, Automation and Robotics (ICINCO 2017)*, Madrid, Spain, July 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01533136

[9] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207.

[10] O. E. Ramos and K. Hauser, "Generalizations of the capture point to nonlinear center of mass paths and uneven terrain," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 851–858.

[11] S. Caron, A. Escande, L. Lanari, and B. Mallein, "Capturability-based Analysis, Optimization and Control of 3D Bipedal Walking," 2018. [Online]. Available: https://arxiv.org/pdf/1801.07022.pdf

[12] A. Del Prete, S. Tonneau, and N. Mansard, "Zero Step Capturability for Legged Robots in Multicontact," *IEEE Transactions on Robotics (TRO)*, vol. 34, no. 4, pp. 1021–1034, 2018.

[13] H. Dai, "Robust multi-contact dynamical motion planning using contact wrench set," Ph.D. dissertation, Massachusetts Institute of Technology, 2016.

[14] S. Caron, Q.-C. Pham, and Y. Nakamura, "Leveraging Cone Double Description for Multi-contact Stability of Humanoids with Applications to Statics and Dynamics," in *RSS*, no. 8, 2015.

[15] R. Orsolino, M. Focchi, C. Mastalli, H.Dai, D. G. Caldwell, and C. Semini, "Application of Wrench based Feasibility Analysis to the Online Trajectory Optimization of Legged Robots," *IEEE Robotics and Automation Letters (RA-L)*, pp. 3363–3370, 2018.

[16] S. Caron, Q.-C. Pham, and Y. Nakamura, "ZMP support areas for multi-contact mobility under frictional constraints," *Transactions on Robotics (TRO)*, vol. 33, pp. 67–80, 2017.

[17] R. Orsolino, M. Focchi, S. Caron, G. Raiola, V. Barasuol, and C. Semini, "Feasible Region: an Actuation-Aware Extension of the Support Region," *IEEE Transactions on Robotics (TRO)*, 2020.

[18] R. J. Griffin, G. Wiedebach, S. McCrory, S. Bertrand, I. Lee, and J. Pratt, "Footstep planning for autonomous walking over rough terrain," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 9–16.

[19] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 279–286.

[20] S. Tonneau, D. Song, P. Fernbach, N. Mansard, M. Tax, and A. Del Prete, "SL1M: Sparse L1-norm Minimization for contact planning on uneven terrain," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 6604–6610.

[21] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernando-Lopez, and C. Semini, "Simultaneous Contact, Gait and Motion Planning for Robust Multi-Legged Locomotion via Mixed-Integer Convex Optimization," in *IEEE Robotics and Automation Letters (RA-L)*, 2018.

[22] 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 (RA-L)*, pp. 1560–1567, 2018.

[23] 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

[24] P. Fernbach, S. Tonneau, O. Stasse, J. Carpentier, and M. Tax, "C-CROC: Continuous and Convex Resolution of Centroidal Dynamic Trajectories for Legged Robots in Multicontact Scenarios," *IEEE Transactions on Robotics (TRO)*, pp. 1–16, 2020.

[25] T. Kwon, Y. Lee, and M. Van De Panne, "Fast and Flexible Multi-legged Locomotion using Learned Centroidal Dynamics," *ACM Transactions on Graphics*, vol. 39, 07 2020.

[26] J. Viereck and L. Righetti, "Learning a Centroidal Motion Planner for Legged Locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, November 2021. [Online]. Available: https://arxiv.org/pdf/2011.02818.pdf

[27] J. Carpentier, R. Budhiraja, and N. Mansard, "Learning Feasibility Constraints for Multi-contact Locomotion of Legged Robots," in *RSS*, 2017.

[28] T. Bretl and S. Lall, "A fast and adaptive test of static equilibrium for legged robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 1109 – 1116.

[29] C. Clark, "Autonomous Robot Locomotion," 2011. [Online]. Available: https://www.cs.princeton.edu/courses/archive/fall11/cos495/COS495-Lecture3-RobotMotion.pdf

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, "Real-Time Trajectory Adaptation for Quadrupedal Locomotion using Deep Reinforcement Learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[32] A. Waetcher, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006. [Online]. Available: https://doi.org/10.1007/s10107-004-0559-y

[33] O. Melon, R. Orsolino, M. Geisert, D. Surovik, I. Havoutis, and M. Fallon, "Receding-Horizon Perceptive Trajectory Optimization for Dynamic Legged Locomotion with Learned Initialization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.