Time-Varying Model Predictive Control for Highly Dynamic Motions of Quadrupedal Robots

Gabriel García[1], Robert Griffin[1], and Jerry Pratt[1]

***Abstract*— Obtaining highly dynamic behaviors in robots is an extremely difficult task. In recent years, sophistication in mechanical design, improved algorithms, and high computational power allows new robots to perform natural gaits and more dynamic behaviors such as backflips. Off-line optimization is necessary to obtain good performance in those difficult tasks. However, when a human performs a backflip or any parkour movements, the computation during those activities is updated online. One of the biggest challenges in robotics is to perform these awesome movements using online optimization with a good understanding of the dynamics of the robot. In this paper we present an approach to deal with complicated dynamic tasks that the world allows the robot to do using online opimization. We aim to perform convex optimization to obtain highly dynamic behaviors from the control point of view.**

Fig. 1. Mini-Cheetah Robot changing its orientation according to the terrain with a rotated friction cone in the MPC and WBC controller

I. INTRODUCTION

Quadruped robots are good platforms for understanding the interaction of robots with the world. All walking robots are underactuated, and that underactuation can be understood using the "centroidal dynamics" [1], which contains the physical concepts of angular and linear momentum. Although robots have many degrees of freedom, quadrupeds usually concentrate its mass and inertia in the body, so they can be studied with a reasonable approximation using the Single Rigid Body Model (SRBM), which contains the centroidal dynamics. In consequence, quadrupeds and their SRBM models are very useful to understand how forces affect the angular and linear momentum of the robot, and in consequence its spatial and rotational movement.

If we desire a robot to avoid breaking its contacts, we know that the the forces and torques affecting the centroidal dynamics must remain inside a 6-D region [2], often called Contact Wrench Cone (CWC). This CWC can be built using the individual Contact Wrench Restrictions, which can have a closed form for specific shapes, such as a rectangular foot [3]. For point contacts, this CWC is reduced to the classical Coulomb friction restrictions, which therefore, model the dynamic stability of a robot with point-foot contact only, such as quadrupeds.

Although methods for the control of robots using the CWC (or reaction forces) had been developed, they always have some limitation when they consider the angular momentum. Direct optimization on the joints can result in highly dynamic behaviors at the cost of offline computation, and slow solving time [4]. Promising approaches include the splitting of the control of the robot in two parts: first the centroidal dynamics, and later the joint control; but even the control of centroidal dynamics is difficult given its non-linear nature. In [5] authors decompose the centroidal dynamics to a convex and a concave part, and they use a QCQP (Quadratically Constrained Quadratic Program) for finding feasible trajectories. [6] consider trajectories kinematically generated and dynamically feasible, but they also use a QCQP for the trajectories. The most important limitation of these approaches is the solving time compared to a common QP (Quadratic Program). In [7] authors solve for the centroidal dynamics using polynomial curves for the Center of Mass (CoM) and the angular momentum. Although this results in continuous trajectories, they have a limitation of generality: the CoM trajectories have a maximum degree, and in consequence the number of parameters is limited. Perhaps a less known fact is that the dynamics of the angular momentum are fully linear after a transformation that computes the angular momentum around a fixed point in the world coordinates. This was used in [8] to obtain trajectories with an upper bound minimized in the angular momentum with a QP program, and later in [9] to minimize the angular momentum in an iterative QP configuration.

After the computation of the trajectories of the centroidal dynamics and reaction forces in the world, the robot must use the joint torques to apply those force in its contacts, and at the same time track the trajectories of the centroidal dynamics and additional tasks [10]. This problem is solved by the Whole-Body Controllers (WBC), well studied in the literature, and solved through QP, fast enough for real applications [11], [12], [13], [14].

When we consider a quadruped robot with massless legs, the angular momentum is very well described by the SRBM,

*This work was supported by IHMC

[1]Authors are with The Florida Institute for Human and Machine Cognition, Pensacola, FL, USA {ggarcia, rgriffin, jpratt}@ihmc.us

which is the model considering the centroidal dynamics plus the orientation of the body. Many techniques used in centroidal dynamics can be applied here, where the concepts of *dynamic feasibility* still exists. In [15] and [16], authors used a simplified version of the SRBM to control the orientation of MIT Mini-Cheetah. This works really well for linear motions, but the simplifications do not allow the robot to perform fast rotational movements or high slope inclinations. In [17], [18] authors use the same simplified model, and they are focused on the tracking of heuristics. In [19] they provide a general method for obtaining heuristics through offline optimization, and online adaptive learning. Although there is not guarantee of dynamic feasibility in those heuristics, this approach seems to be promising to increase of the capabilities of movements in the robot.

A. Contributions

The main contribution of this paper is stated as follows: Design an iterative QP Optimization that produces, from higher descriptions of a highly dynamic behavior, dynamically feasible trajectories for the centroidal dynamics of a quadruped, and quasi-feasible trajectories in the orientation of its body considering the Single Rigid Body Model in any orientation of the terrain, when the footsteps and the timing between steps are fixed.

By *quasi-feasible*, we refer that the real trajectory is approximated with an error of $O(\|\mathbf{x}-\mathbf{x}^*(t)\|^2)$, where $\mathbf{x}(t)$ are the states of the system, and $\mathbf{x}^*(t)$ are the predicted states. Specifically, in [14], authors mention that the following simplifications are done in order to perform the convex MPC from [16]:

- The roll and pitch angles are small.
- States are close to the commanded trajectory: A time-varying linearization is performed.
- Pitch and roll velocities and off-diagonal terms of the inertia tensor are small.

The specific contributions in this paper are:

- Remove the first assumption, thus, the robot can be oriented in any direction.
- Remove the third assumption, so fast rotational motions can be achievable.

Combining these contributions, we are able to generate more dynamic movements, such as walking in any orientation, walking on a V-shaped terrain, and leaping over a sloped terrain.

II. BACKGROUND

A. Mathematical Model

The centroidal dynamics of any Multi-body System under multi-contact, point-foot placements holds as:

$$\begin{bmatrix} \ddot{\mathbf{r}} = \sum_{i=1}^{n}\mathbf{f}_i + \mathbf{g} \\ \dot{\mathbf{L}} = \sum_{i=1}^{n}(\mathbf{r}_{pi}-\mathbf{r})\times\mathbf{f}_i \end{bmatrix} \tag{1}$$

Where $\mathbf{r}$ is the position of the CoM, $\mathbf{f}_i$ is the vectorial i-th ground reaction force, $\mathbf{r}_{pi}$ is the position of the i-th contact, m is the mass of the robot and $\mathbf{g}$ is the gravity vector. We will be assuming $\mathbf{r}_{pi}$ already given in time.

When the system has only one link with mass (i.e. body, the legs are massless), the angular momentum $\mathbf{L}$ is reduced to

$$\mathbf{L} = \mathbf{I}\omega = \mathbf{R}\mathbf{I_B}\mathbf{R^T}\omega \tag{2}$$

where $\mathbf{I_B}$ represents the Inertia tensor of the body in a local frame, $\mathbf{I}$ represents the Inertia in the world coordinates, $\mathbf{R}$ is the rotation matrix from the local to the world coordinates of the body, and ω represents the angular velocity of the body. The dynamics of $\mathbf{R}$ hold:

$$\dot{\mathbf{R}} = [\omega]_\times \mathbf{R} \tag{3}$$

where $[\bullet]_\times$ denotes the matrix form of the cross product. If we combine (3) with (1), we obtain the dynamics of the SRBM:

$$\begin{bmatrix} \ddot{\mathbf{r}} = \sum_{i=1}^{n}\mathbf{f}_i + \mathbf{g} \\ \dot{\mathbf{L}} = \sum_{i=1}^{n}(\mathbf{r}_{pi}-\mathbf{r})\times\mathbf{f}_i \\ \dot{\mathbf{R}} = [\omega]_\times \mathbf{R} \end{bmatrix} \tag{4}$$

The control of the orientation matrix $\mathbf{R}$ is known as the *attitude control problem*. In [16], Euler angles are used to represent the rotation matrix, which unfortunately has a singularity when the roll is $\pm\frac{\pi}{2}$, which will forbid the robot to perform extreme behaviors such as parkour wall-jumps or climbs. In this research, we use the well-known approach of the quaternion representation $\mathbf{q}_r$, which uses 4 variables and does not have a singularity. Let $\mathbf{q}_r$ be the quaternion representing the orientation of the robot with respect to a fixed coordinate system in the world frame. The dynamics of $\mathbf{q}_r$ are

$$\dot{\mathbf{q}}_r = \frac{1}{2}\mathbf{q}_r \circ \omega = \mathbf{Q}(\mathbf{q}_r)\omega \tag{5}$$

where $\circ$ is the quaternion product and $\mathbf{Q}(\mathbf{q}_r) \in \mathbb{R}^{4\times 3}$ is its matrix form representation.

We do not simplify $\frac{d}{dt}(\mathbf{I}\omega) = \mathbf{I}\dot{\omega} + \omega \times (\mathbf{I}\omega) \approx \mathbf{I}\dot{\omega}$, as is usually done. Instead, we will use the angular momentum $\mathbf{L}$ as our state variable instead of ω leading to:

$$\mathbf{L} = \mathbf{I}\omega \rightarrow \omega = \mathbf{I^{-1}}\mathbf{L} \tag{6}$$

The inertia $\mathbf{I}$ is a function of the orientation of the robot, specifically $\mathbf{I} = \mathbf{R}\mathbf{I_B}\mathbf{R^T}$. The rotation $\mathbf{R}$ is a function of the quaternion $\mathbf{q}_r$, so the inertia $\mathbf{I}$ is completely determined by the quaternion $\mathbf{q}_r$ as follows (the matrix $\mathbf{I}_B$ is constant):

$$\mathbf{I}(\mathbf{q}_r) = \mathbf{R}(\mathbf{q}_r)\mathbf{I_B}\mathbf{R}(\mathbf{q}_r)^T \tag{7}$$

Using (7) and replacing (6) in (5) we obtain the dynamics of $\mathbf{q}_r$:

$$\dot{\mathbf{q}}_r = \mathbf{Q}(\mathbf{q}_r)\mathbf{I^{-1}}(\mathbf{q_r})\mathbf{L} \tag{8}$$

We are using the angular momentum $\mathbf{L_2}$ around the origin of the world coordinates instead of the angular momentum around the CoM. This linearizes the second equation of (1) [4], [9].

$$\mathbf{L}_2 = \mathbf{L} + m\mathbf{r}\times\dot{\mathbf{r}} \tag{9}$$

Taking time derivative to (9) and using (1) we obtain:

$$\dot{\mathbf{L}}_2 = \sum_{i=1}^{n} \mathbf{r}_{pi} \times \mathbf{f}_i + m\mathbf{r} \times \mathbf{g} \tag{10}$$

Now using the dynamics of $\mathbf{q}_r$ (8) and $\mathbf{L_2}$ (10) we obtain another representation of the dynamics of the SRBM, over which we can apply a time-varying linearization *only in the orientation* and apply QP strategies to obtain forces which later can be tracked by a WBC:

$$\begin{bmatrix} \ddot{\mathbf{r}} = \sum_{i=1}^{n} \mathbf{f}_i + \mathbf{g} \\ \dot{\mathbf{L}}_2 = \sum_{i=1}^{n} \mathbf{r}_{pi} \times \mathbf{f}_i + m\mathbf{r} \times \mathbf{g} \\ \dot{\mathbf{q}}_r = \mathbf{Q}(\mathbf{q}_r)\mathbf{I}^{-1}(\mathbf{q_r})(\mathbf{L}_2 - m\mathbf{r} \times \dot{\mathbf{r}}) \end{bmatrix} \tag{11}$$

We define $\mathbf{F(x)} = \mathbf{Q}(\mathbf{q}_r)\mathbf{I}^{-1}(\mathbf{q_r})(\mathbf{L}_2 - m\mathbf{r} \times \dot{\mathbf{r}})$ where $\mathbf{x} = [\mathbf{r}, \dot{\mathbf{r}}, \mathbf{L}_2, \mathbf{q}_r]^T$ are the states of the system, so we have $\dot{\mathbf{q}}_r = \mathbf{F}(\mathbf{x})$. In this equation we apply a time-varying linearization using the Taylor expansion around a reference trajectory $\mathbf{x}^*(t)$ for the states:

$$\dot{\mathbf{q}}_r = \mathbf{F}(\mathbf{x}^*(t)) + \left.\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*(t)} (\mathbf{x} - \mathbf{x}^*(t)) + O(\|\mathbf{x} - \mathbf{x}^*(t)\|^2) \tag{12}$$

We have now the linearized system of the SRBM with point-foot contact:

$$\begin{bmatrix} \ddot{\mathbf{r}} = \sum_{i=1}^{n} \mathbf{f}_i + \mathbf{g} \\ \dot{\mathbf{L}}_2 = \sum_{i=1}^{n} \mathbf{r}_{pi} \times \mathbf{f}_i + m\mathbf{r} \times \mathbf{g} \\ \dot{\mathbf{q}}_r = \mathbf{F}(\mathbf{x}^*) + \frac{\partial \mathbf{F}}{\partial \mathbf{x}}\big|_{\mathbf{x}=\mathbf{x}^*} (\mathbf{x} - \mathbf{x}^*) \end{bmatrix} \tag{13}$$

The linearization done in (12) has errors of order $O(\|\mathbf{x} - \mathbf{x}^*(t)\|^2)$. As long as the states $\mathbf{x}$ stay close to $\mathbf{x}^*(t)$, these errors will be considerably small and the dynamics will be dominated by the term $(\mathbf{x} - \mathbf{x}^*(t))$, which are taken into account in (13).

After this, the application of the control is the same as the MPC of (18) from [16], a quadratic optimization is performed to obtain the reaction forces that track the desired trajectory, followed by a WBC to track these forces. System (13) only keeps the second simplification of [2]: The states are close to the commanded trajectories, but this time the body is allowed to have any orientation without singularities and it can be spinning at high angular velocities.

A minor consideration is that we are going to add friction restrictions according to the orientation of the terrain. The inequalities in flat ground are:

$$\begin{matrix} -\mu f_z \leq f_x \leq \mu f_z \\ -\mu f_z \leq f_y \leq \mu f_z \\ 0 \leq f_z \leq f_{zmax} \end{matrix} \tag{14}$$

These inequalities are condensed in matrix form as:

$$\mathbf{W}_i \mathbf{f}_i \leq \mathbf{b}_i \tag{15}$$

Where $\mathbf{W}_i \in \mathbb{R}^{6\times 3}$, $\mathbf{b}_i \in \mathbb{R}^{6\times 1}$ and $\mathbf{f}_i$ contains the stacked forces in x, y, and z. This is done for each i-th contact.

In a non-horizontal plane, the linearized friction cone is $\mathbf{W}_i$ rotated with the transpose of the matrix $\mathbf{R}_{plai}$ mapping from the world to the plane where the i-th contact is:

$$\mathbf{W_i}\mathbf{R}_{plai}^T\mathbf{f_i} \leq \mathbf{b_i} \tag{16}$$

III. CONTROL APPROACH

Most of the control approach is taken from [15], [14] and [16], with small modifications in the algorithm and the base code for an improved performance. The main contribution of this paper is Subsection B.

A. Contact Scheduling

Finding trajectories for the centroidal dynamics of robots while considering a time variable for the footsteps is a non-linear problem [8], [22]. Although it can be handled under certain relaxations such as constant angular momentum [22], for general motions the problem disappears when we consider footsteps timings already fixed [4], [7], [9], [15].

The contact schedule is taken from [15] and [17]. The gait uses time-based phases for the contact scheduling. They are periodic and the legs of the robot are enforced to touch the ground when they are estimated to be in contact with the ground according to the gait scheduling. This is always possible under the assumption of massless legs (and also assuming that the contact position is kinematically feasible, i.e., inside the leg range limits).

Changes on the contact schedule include a concatenation in the footsteps between different behaviors. This is done for feasibility purposes. Although the jumps are defined as gaits, they are cut before they start their periodic repetition, and instead, the contact placements of the next gait are used. This already provides a trajectory dynamically feasible to the next gait for tracking. See the next subsection for more details.

B. MPC Optimization

The planning for feasible trajectories is one of the biggest challenge in robotics, specially because of the nonlinearity in the SRBM in (11). Although for common locomotion the problem is somehow addressed, highly dynamic motions with online computation is still a very open research problem. This subsection provide an alternative approach for solving those problems and capable of more dynamic movements than previous approaches.

We are going to perform an initial QP optimization using a heuristic $\mathbf{x_H}^*$ for the initial base trajectory, and we are going to perform successive QP optimizations on the linearized system (13) around the resulting trajectories. We also minimize the distance between the future trajectories and the previous resulting trajectories, making this is a kind of nonlinear MPC controller.

The initial heuristic models the behavior of the robot in the future and provides a high level description of the position and orientation of the robot, but it does not need to be perfectly dynamically feasible. The results of predictive controllers based on heuristics for quadrupeds [17], [18], [19] highly rely on those predefined heuristics (the robot always tries to track those heuristics), so the more dynamically feasible they are, the better. In this work, the initial heuristic is only a guess of how the robot will move given a desire behavior, so it's just an initialization. After the first optimization is solved, the robot will track the resulting improved trajectory.

The iterative MPC setup is defined as $\mathbb{P}_k$:

$$\mathbb{P}_k : p_k^* = \min_{\mathbf{x},\mathbf{u}} \int_0^{t_f} \|\mathbf{x}_{k-1}^*(t) - \mathbf{x}\|_{\mathbf{L}} + \|\mathbf{u}\|_{\mathbf{K}} dt, \ \textbf{s.t.} \tag{17}$$
$$\dot{\mathbf{x}} = A_{k-1}(t)\mathbf{x} + B_{k-1}(t)\mathbf{u} + C_{k-1}(t)$$
$$\mathbf{x}(0) = \mathbf{x}_0$$
$$W(t) R_{pla}^T(t)\mathbf{u} \leq b(t)$$

Where $W(t)$, $R_{pla}(t)$ and $b(t)$ encodes the friction restrictions defined (16) and $\mathbf{u}$ contains the reaction forces $\mathbf{f}_i$ stacked vertically.

We define $\mathbf{x}_{k,pre}^*(t)$ as the resulting trajectory of the optimization $\mathbb{P}_k$:

$$\mathbf{x}_{k,pre}^*(t) = \underset{\mathbf{x}}{\operatorname{argmin}} \, \mathbb{P}_k \tag{18}$$

The update rule in $\mathbf{x}_k(t)$, defined in $[0, t_f]$ is given by:

$$\mathbf{x}_k^*(t) = \begin{cases} \mathbf{x}_{k,pre}^*(t + \Delta t_{MPC}) & \forall t \in [0, t_f - \Delta t_{MPC}] \\ \mathbf{x}_{d,H}^*(t) & \forall t \in [t_f - \Delta t_{MPC}, t_f] \end{cases} \tag{19}$$

We are essentially shifting the new trajectory by a time of Δt_{MPC}, which corresponds to the time when each optimization is solved. The last section of the trajectory given by $\mathbf{x}_{d,H}^*(t)$ corresponds to an heuristic for the new desired states provided by a higher level input. See Section IV for more details. We can obtain the expressions for $A_{k-1}(t)$, $B_{k-1}(t)$ and $C_{k-1}(t)$ from (13):

$$A_{k-1}(t) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} \\ -m[\mathbf{g}]_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} \\ \frac{\partial \mathbf{F}}{\partial \mathbf{r}}\big|_{\mathbf{x}_{k-1}^*} & \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{r}}}\big|_{\mathbf{x}_{k-1}^*} & \frac{\partial \mathbf{F}}{\partial \mathbf{L_2}}\big|_{\mathbf{x}_{k-1}^*} & \frac{\partial \mathbf{F}}{\partial \mathbf{q_r}}\big|_{\mathbf{x}_{k-1}^*} \end{bmatrix} \tag{20}$$

$$B_{k-1}(t) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{I}_{3\times3} \\ [\mathbf{r}_{p1}]_\times & [\mathbf{r}_{p2}]_\times & [\mathbf{r}_{p3}]_\times & [\mathbf{r}_{p4}]_\times \\ \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} \end{bmatrix} \tag{21}$$

$$C_{k-1}(t) = \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{g} \\ \mathbf{0}_{3\times1} \\ \mathbf{F}(\mathbf{x}_{k-1}^*) - \frac{\partial \mathbf{F}}{\partial \mathbf{x}}\big|_{\mathbf{x}_{k-1}^*} \mathbf{x}_{k-1}^* \end{bmatrix} \tag{22}$$

In general, the matrix B_{k-1} depends on the footstep placement. Given the assumption that we know the foot placements and their time from the gait schedule, B_{k-1} only depends on time. Matrix A_{k-1} and C_{k-1} contains the *exact* dynamics of the CoM position, velocity and angular momentum, and the linearization of the dynamics of the orientation $\mathbf{q}_r$ from (12) with an error of $O(\|\mathbf{x} - \mathbf{x}_{k-1}^*\|^2)$.

C. QP optimization

The MPC setup defined in (17) is now written in discrete time as a QP optimization using the direct transcription method with a condensed approach. The problem is solved as [16], with minor modifications to include the term C_{k+1}.

The dynamics in (17) are discretized as:

$$\mathbf{x}[n+1] = \mathbf{A}_{k-1}[n]\mathbf{x}[n] + \mathbf{B}_{k-1}[n]\mathbf{u}[n] + \mathbf{C}_{k-1}[n] \tag{23}$$

Solving for each $\mathbf{x}[n]$, we can condense the equations to:

$$\mathbf{X} = \mathbf{A}_{qp}\mathbf{x}_0 + \mathbf{B}_{qp}\mathbf{U} + \mathbf{C}_{qp} \tag{24}$$

Note that we are omitting the subindex $k-1$ in $\mathbf{A}_{qp}$, $\mathbf{B}_{qp}$, and $\mathbf{C}_{qp}$ (They are built using $\mathbf{A}_{k-1}$, $\mathbf{B}_{k-1}$ and $\mathbf{C}_{k-1}$). The cost function in (17) corresponds to

$$J(\mathbf{U}) = \|\mathbf{A}_{qp}\mathbf{x}_0 + \mathbf{B}_{qp}\mathbf{U} + \mathbf{C}_{qp} - \mathbf{x}_{k-1}^*\|_{\mathbf{L}} + \|\mathbf{U}\|_{\mathbf{K}} \tag{25}$$

We have the QP optimization written as

$$\mathbb{P}_{\mathbb{D}k} : p_k^* = \min_{\mathbf{U}} \frac{1}{2}\mathbf{U^T H_{k-1} U} + \mathbf{U^T g_{k-1}}, \ s.t. \tag{26}$$
$$\mathbf{W_{qp} U} \leq \mathbf{b_{qp}}$$

where $\mathbf{W_{qp}}$ and $\mathbf{b_{qp}}$ (also dependent on k) contains the restrictions of (17) related with the friction constraints and:

$$\mathbf{H}_{k-1} = \mathbf{2}(\mathbf{B_{qp}^T L B_{qp}} + \mathbf{K}) \tag{27}$$

$$\mathbf{g}_{k-1} = \mathbf{2B_{qp}^T L}(\mathbf{A_{qp}x}_0 + \mathbf{C_{qp}} - \mathbf{x}_{k-1}^*) \tag{28}$$

D. Whole-Body Controller

The iterative QP formulation $\mathbb{P}_{\mathbb{D}k}$ given in (26) finds desired forces to apply to the the world. The robot applies torques to the joint, and this is traduced into forces to the world. The solution for that problem is a WBC given in [14]. The system is modeled as a Floating Base Model and a QP formulation solves for reaction forces and floating base accelerations. We use the same approach with a minor change in the force restrictions to allow the robot to use any orientation in the foot, similar to the force restriction in (17) and (16) as follows:

$$\mathbf{W R}_{pl}^T \mathbf{f_r} \leq \mathbf{b} \tag{29}$$

where $\mathbf{W}$ and $\mathbf{b}$ contains the friction restrictions for horizontal terrain, $\mathbf{R}_{pl}$ is the rotation from the world to the plane orientation and $\mathbf{f}_r$ contains the forces of each contact vertically stacked.

IV. DYNAMIC MOTIONS

This subsection provide an approach for solving the problem of highly dynamic motions, specifically:

- Walk and jump over sloped terrain.
- Walk and jump over V-shaped terrain.
- Leaps over sloped terrain.

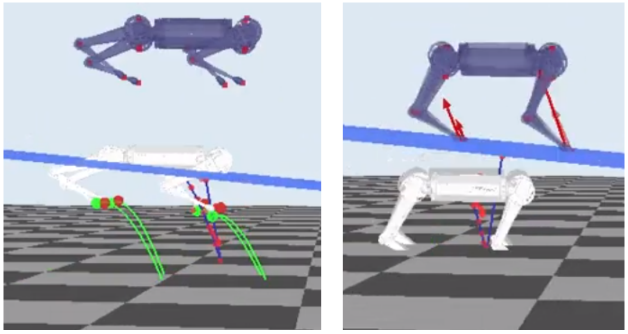

Fig. 2. Mini-Cheetah Sim jumping over sloped terrain. (a) Aerial phase of the real robot (blue) (b) Forces are applied according to the orientation of the terrain to stop the velocity gained in the aerial phase

A. Sloped terrain

In normal locomotion, a higher level input can provide a commanded velocity and rotation to the robot. Those values are inserted in the QP as the heuristic $\mathbf{x}_{d,H}^*(t)$ in (19):

$$\mathbf{x}_{d,H}^*(t) = \begin{bmatrix} \mathbf{r}_0 + t\mathbf{v}_{cmd} \\ \mathbf{v}_{cmd} \\ m\mathbf{r}_0 \times \mathbf{v}_{cmd} \\ E2q_r(\Theta_0 + t\dot{\Theta}_{cmd}) \end{bmatrix} \tag{30}$$

where $\mathbf{r}_0$ is the current position, $\mathbf{v}_{cmd}$ is the commanded velocity, Θ_0 is the current orientation of the robot in Euler angles, $\dot{\Theta}_{cmd}$ is the commanded angular velocity of the robot in Euler angles (e,g. roll and yaw derivative) and $E2q_r$ is a mapping from Euler angles to quaternions.

In sloped terrain, we desired to orient the robot according to the terrain, so in this case, we are going to rotate the desired orientation using the rotation matrix of the plane $\mathbf{R}_p$ giving:

$$\mathbf{x}_{d,H}^*(t) = \begin{bmatrix} \mathbf{r}_0 + t\mathbf{v}_{cmd} \\ \mathbf{v}_{cmd} \\ m\mathbf{r}_0 \times \mathbf{v}_{cmd} \\ RM2q_r(E2RM(\Theta_0 + t\dot{\Theta}_{cmd})\mathbf{R}_p^T) \end{bmatrix} \tag{31}$$

where $E2RM$ is a mapping from Euler angles to its rotation matrix, and $RM2q_r$ is a mapping from the rotation matrices to its quaternion equivalent.

Although a better heuristic can be implemented considering the angular momentum as a function of $\dot{\Theta}_{cmd}$ and Θ_0, the current heuristic enforces an angular momentum around the CoM of $\mathbf{0}$, which is usual in normal locomotion.

The matrix $\mathbf{R}_p$ can be obtained by prior knowledge of the terrain, or detected by the robot using a regression with minimum squares in the estimated contact placements as is described in [20]. The matrix $\mathbf{R}_{pl}$ from (29) can be used as the matrix $\mathbf{R}_p$ stacked diagonally, as well as the matrix $\mathbf{R}_{pla}$ from (17) for the QP formulation.

B. V-shaped terrain

In [21] authors presented the robot HyQ performing a walk in a V-shaped terrain where there is not enough friction to apply a vertical force. The robot uses quasi-static postures for its movements. In this subsection, we apply our control approach to a dynamically walk over a V-shaped terrain.

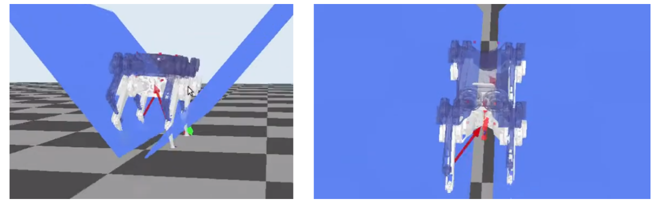

Fig. 3. Mini-Cheetah Sim walking over V-shaped terrain. Forces cannot be applied vertically (red arrows)

This case is very similar to the walking over sloped terrain, but with some changes in the desired heuristic and the rotation matrices in the WBC and the MPC. We are going to use the heuristic defined in (30), because we want the base orientation of the robot to be horizontal. We also assume knowledge of the terrain, specifically, the orientation of each side of the terrain.

The matrix $\mathbf{R}_{pl}$ from (29) are defined as the rotation matrices of the plane where belongs each foot. Let $\mathbf{R}_{V1}$ and $\mathbf{R}_{V2}$ be the rotation matrices of each side of the V-shaped terrain. Then, the matrix $\mathbf{R}_{pl}$ is defined as:

$$\mathbf{R}_{pl} = \mathrm{diag}([\mathbf{R}_{V1}, \mathbf{R}_{V2}, \mathbf{R}_{V2}, \mathbf{R}_{V1}]) \tag{32}$$

where *diag* is the diagonal operator. Meanwhile, the matrix $\mathbf{R}_{pla}$(t) from (17) is defined as $\mathbf{R}_{pl}$, as long as the robot is staying on the V-shaped terrain at the time t.

C. Leap

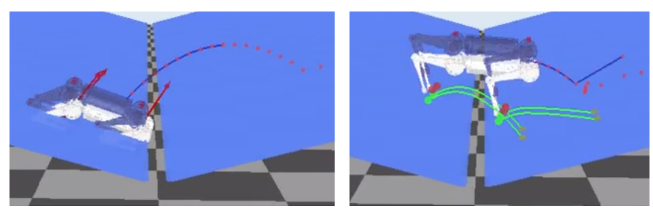

Fig. 4. Mini-Cheetah Sim performing a leap between two planes. (a) Initial forces and trajectory (b) Flight phase and future positions of the CoM according to the MPC

In order to obtain a leap over a sloped terrain, we assume again that the terrain is already known. In this case, when the leap event is triggered, the optimization $\mathbb{P}_k$ is restarted, i.e., run from $k = 1$. Here, the initial trajectory to track $\mathbf{x}_0^*$ is defined as:

$$\mathbf{x}_0^*(t) = \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{q}_{r0} \end{bmatrix} \forall t \in [0, t_{LO}] \tag{33}$$

$$\mathbf{x}_0^*(t) = \begin{bmatrix} \mathbf{r}_0 + \mathbf{v_H}(t - t_{LO}) + \frac{1}{2}\mathbf{g}(t - t_{LO})^2 \\ \mathbf{v_H} + \mathbf{g}(t - t_{LO}) \\ \mathbf{L_{2H}}(t) \\ \frac{\mathbf{q}_{r0}(t_{la} - t) + \mathbf{q}_{rf}(t - t_{LO})}{\|\mathbf{q}_{r0}(t_{la} - t) + \mathbf{q}_{rf}(t - t_{LO})\|} \end{bmatrix} \forall t \in [t_{LO}, t_{la}] \tag{34}$$

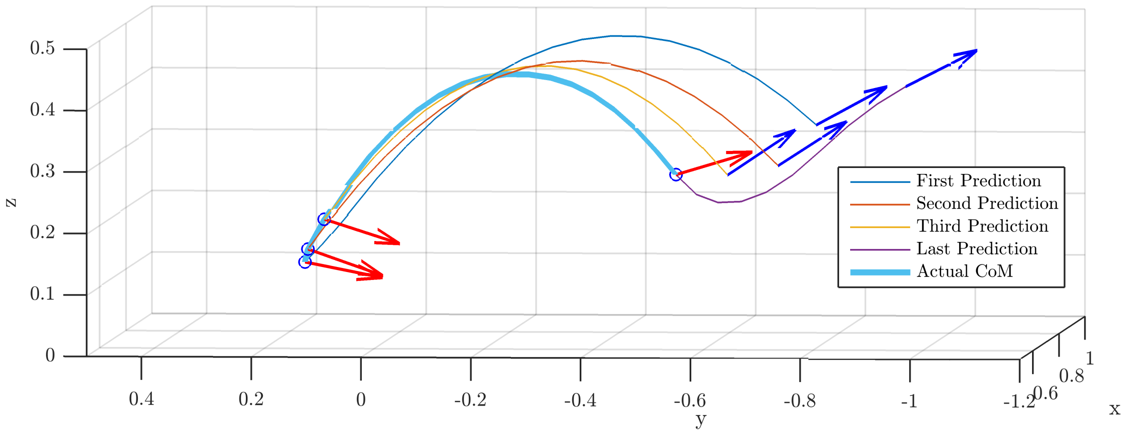

Fig. 5. Actual and predicted Mini-Cheetah CoM trajectory of a leap exported from the MIT Cheetah physics simulator. Orientation (roll) plotted as arrows. Red arrows represent the initial orientation. Blue arrows represent the predicted final orientation.

$$\mathbf{x}_0^*(t) = \begin{bmatrix} \mathbf{r}_f \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{q}_{rf} \end{bmatrix} \forall t \in [t_{la}, t_f] \tag{35}$$

where t_{LO} represents the lift-off time, t_{la} the landing time, r_0 the initial position, $\mathbf{q}_{r0}$ the initial orientation (quaternion), r_f the final desired position, $\mathbf{q}_{rf}$ the final desired orientation, and:

$$\mathbf{v_H} = \frac{\mathbf{r}_f - \mathbf{r}_0 - \frac{1}{2}\mathbf{g}(t_{la} - t_{LO})^2}{t_{la} - t_{LO}} \tag{36}$$

$$\mathbf{L_{2H}}(t + t_{LO}) = \mathbf{L_H} + m(\mathbf{r}_0 - \frac{1}{2}\mathbf{g}t^2) \times (\mathbf{v_H} + \mathbf{g}t) \tag{37}$$

$$\mathbf{L_H} = \mathbf{M}^{\dagger}\left(\frac{\mathbf{q}_{r0} + \mathbf{q}_{rf}}{\|\mathbf{q}_{r0} + \mathbf{q}_{rf}\|}\right)\frac{\mathbf{q}_{rf} - \mathbf{q}_{r0}}{t_{la} - t_{LO}} \tag{38}$$

with $\mathbf{M}^{\dagger}(\mathbf{q}_r)$ denotes the pseudoinverse of $\mathbf{M}(\mathbf{q}_r) = \mathbf{Q}(\mathbf{q}_r)\mathbf{I}^{-1}(\mathbf{q}_r)$ from the last equation of (13).

The heuristic essentially tells the robot to stay in place as much as it can before lifting-off in (33), then the robot tries to follow a ballistic trajectory that lands in the final desired position according to (34) and (36). The robot also tries to reach a constant angular momentum (38) around its CoM that roughly approximates the angular momentum required to go from the orientation $\mathbf{q}_{r0}$ to $\mathbf{q}_{rf}$ under torque-free conditions (Finding that real constant angular momentum is a non-trivial problem). The derivation of these initial trajectory to track can be tedious and it is outside the scope of the paper.

Now, the first QP program is solved, ideally $\mathbb{P}_0$ (17), but it is approximated by $\mathbb{P}_{\mathbb{D}0}$ (26). Although $\mathbf{x}_0^*(t)$ is not dynamically feasible, it contains enough information to allow $\mathbb{P}_{\mathbb{D}0}$ to generate a trajectory $\mathbf{x}_{1,pre}^*(t)$ that describes what are we desiring to do, i.e. leap from an initial position and orientation to a different position and orientation while following a flight phase. $\mathbf{x}_{1,pre}^*(t)$ is now *quasi-dynamically feasible*, i.e. it is feasible in the components $\mathbf{r}(t), \dot{\mathbf{r}}(t)$ and $\mathbf{L}(t)$, and the orientation $\mathbf{q}_r(t)$ is approximated by a linearization with error $O(\|\mathbf{x}_1^*(t) - \mathbf{x}_0^*(t)\|^2)$. We generate $\mathbf{x}_1^*(t)$ using the update rule from (19). The Heuristic for the end of the trajectory $\mathbf{x}_{d,H}^*(t)$ can be simply (35), i.e. constant final desired position and orientation, so this will work as a landing controller. Another possible heuristic is (31), which can allow even more dynamic behaviors (i.e. Leap and continuous running).

V. DISCUSSION

One limitation of the current approach is that the QP program we are using is a discretization of the continuous optimization, so we have numerical errors. Theoretically this is solved by decreasing the discretization time, but the solving time of the QP optimization grows quadratically with the number of discrete variables (depending on the solver). This is the main limitation which avoid us to perform long-horizon problems, e.g. double or triple continuous leaps coupled with running. The condensed approach for the QP formulation is faster than the other classical approaches, such as sparse direct transcription, but algorithms considering splines may perform better computationally. The solving time problem can be mitigated with a one or two seconds of pre-computation of the desired trajectory for a double leap, which is enough for obtaining a feasible trajectory to track in a reasonable horizon.

Another limitation are the timing and the stepping problem, which we are assuming to be fixed. The stepping problem is very-well addressed for flat terrain (or even for small slopes in the terrain when the steps are projected) by the heuristics described in [15], [20] and [18], but these heuristics do not work in highly inclined terrain or in vertical walls. Finding better heuristics for stepping in complex terrain can be an interesting topic of research. Another interesting topic of study can be the inclusion of the stepping places in the optimization, but this will induce non-linearities in the dynamics, specifically in the matrix $B_{k-1}(t)$ from the dynamics (17), which is no longer time-varying, but it is state dependent. Mixed-Integer Programming can also be a potential candidate for the solution, but the slow solving time is also a problem.

The fixed time between steps is also an issue. The the heuristic **vH** (36) partially mitigates the problem, but we still need to provide good values for the lifting-off and the landing time. Although the optimization in (17) find good corrections for the errors in the dynamics, it is still subject to this problem (times are fixed in the optimization). Although heuristics can also help in the solution of the problem, we wish to obtain online optimization which also modifies the timing between steps. The straightforward way to solve this (implementing a variable time) changes the problem to a complex state-dependent hybrid system, meanwhile giving the program multiple possible times for the solution results again in Mixed-Integer Programming, for which many integer variables slows considerably the optimization.

Additional minor considerations include the knowledge of the terrain. Slopes are solved with perception, but parameters like the friction coefficient are not. We can still obtain good motions using a conservative friction coefficient, but extreme maneuvers in parkour or even simpler movements like the "moonwalk" requires a very good approximation of the friction coefficient.

VI. CONCLUSIONS

In the present work, we obtained highly dynamic feasible trajectories for the centroidal dynamics, and quasi-feasible orientations of the robot using a MPC formulation in a time-varying linearization of the dynamics of the Single Rigid Body Model, which leaves the errors in quadratic terms. The theoretical derivation and the good results obtained in simulation are a solid base for its implementation of in real hardware, from which so far the rotation of the friction cone and the flat terrain behavior had been tested. We are aiming to improve the state estimation to allow the robot to perform those highly dynamic motions.

REFERENCES

[1] David E. Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, (September 2012):1–16, jun 2013.

[2] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa, "A universal stability criterion of the foot contact of legged robots-adios zmp," in *Robotics and Automation (ICRA), 2006 IEEE International Conference on*, 2006, pp. 1976–1983.

[3] S. Caron, Q.-C. Pham, and Y. Nakamura, "Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 2015.

[4] H. Dai, A. Valenzuela, and R. Tedrake. "Whole-Body Motion Planning with Centroidal Dynamics and Full Kinematics." in *Humanoid Robots (Humanoids), 2014 IEEE-RAS 14th International Conference on*, 2014

[5] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, 2016.

[6] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ Int. Conference on*, 2016, pp. 2703–2710.

[7] P. Fernbach, S. Tonneau, and M. Tax, "CROC: Convex resolution of Centroidal Dynamics trajectories to provide a feasibility criterion for the multi contact planning problem," in *Intelligent Robots and Systems (IROS), 2018 IEEE/RSJ International Conference on*, Oct 2018.

[8] H. Dai and R. Tedrake. "Planning robust walking motion on uneven terrain via convex optimization", in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, 2016

[9] G. Garcia, R. Griffin, and J. Pratt "Convex Optimization of the Full Centroidal Dynamics for Planning in Multi-Contact Scenarios," available online at *ResearchGate Preprint*, Dec 2019

[10] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa. "Resolved momentum control: humanoid motion planning based on the linear and angular momentum". In *Intelligent Robots and Systems (IROS) 2003, IEEE/RSJ International Conference on* , 2003

[11] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, pp. 429–455, Mar 2016.

[12] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[13] S. Fahmi, C. Mastalli, M. Focchi and C. Semini, "Passive Whole-Body Control for Quadruped Robots: Experimental Validation Over Challenging Terrain," in *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2553-2560, July 2019.

[14] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim "Highly Dynamic Quadruped Locomotion via Whole-Body Impulse Control and Model Predictive Control," in *arXiv preprint arXiv:1909.0658*, Sep 2019

[15] B. Katz, J. Di Carlo, and S. Kim. "Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control", in *Robotics and Automation (ICRA), 2019 IEEE-RAS International Conference on*, 2019

[16] J. Di Carlo, P. Wensing, B. Katz, G. Bledt, and S. Kim. "Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control", in *Intelligent Robots and Systems (IROS), 2018 IEEE/RSJ International Conference on*, 2018

[17] G. Bledt, P. Wensing and S. Kim. "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the MIT cheetah", in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, 2017

[18] G. Bledt, and S. Kim. "Implementing Regularized Predictive Control for Simultaneous Real-Time Footstep and Ground Reaction Force Optimization", in *Intelligent Robots and Systems (IROS), 2019 IEEE/RSJ International Conference on*, 2019

[19] G. Bledt, and S. Kim. "Extracting Legged Locomotion Heuristics with Regularized Predictive Control", to Appear in *Robotics and Automation (ICRA), 2020 IEEE-RAS International Conference on*, 2020

[20] G. Bledt, P. Wensing and S. Kim. "MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot", in *Intelligent Robots and Systems (IROS), 2018 IEEE-RAS International Conference on*, 2018

[21] M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," *Auton. Robots*, vol. 41, no. 1, pp. 259–272, 2017.

[22] S. Caron, and Q. Pham. "When to make a step? Tackling the timing problem in multi-contact locomotion by TOPP-MPC", in *Humanoid Robots (Humanoids), 2017 IEEE-RAS 17th International Conference on*, 2017