Optimization-free Ground Contact Force Constraint Satisfaction in Quadrupedal Locomotion

Eric Sihite, Yichu Yang, and Alireza Ramezani, *SiliconSynapse Laboratory*[1]

***Abstract*— We are seeking control design paradigms for legged systems that allow bypassing costly algorithms that depend on heavy on-board computers widely used in these systems and yet being able to match what they can do by using less expensive optimization-free frameworks. In this work, we present our preliminary results in modeling and control design of a quadrupedal robot called *Husky Carbon*, which under development at Northeastern University (NU) in Boston. In our approach, we utilized a supervisory controller and an Explicit Reference Governor (ERG) to enforce ground reaction force constraints. These constrained are usually enforced using costly optimizations. However, in this work, the ERG manipulates the state references applied to the supervisory controller to enforce the ground contact constraints through an update law based on Lyapunov stability arguments. As a result, the approach is much faster to compute than the widely used optimization-based methods.**

I. Introduction

In this work, we will report our preliminary results in the modeling, control design and development of a legged robot called *NU's Husky Carbon* (shown in Fig. 1). Our objective is to integrate legged and aerial mobility in a single platform.

Robotic biomimicry of animals multi-modal locomotion can be a significant ordeal. The prohibitive design restrictions such as tight power budget, limited payload, complex multi-modal actuation and perception, excessive number of active and passive joints involved in each mode, sophisticated control and environment-specific models, just to name a few, have alienated these concepts. It is worth noting that as how multi-modality has secured birds' (and other species) survival in complex environments, similar manifestations in mobile robots can be rewarding and important.

Payload is a key factor for aerial systems. We are seeking control design paradigms that allow bypassing costly algorithms that depend on heavy on-board computers widely used in legged systems and yet being able to match what they can do using less expensive optimization-free frameworks. In this work, we will present our optimization-free approach based on path planning in the state space of the robot. Our constraints are bounded unilateral ground contact forces which are important to avoid the violation of gait feasibility conditions. The resulting algorithms can run faster and can be hosted by lightweight processors.

From a feedback design standpoint, the challenge of simultaneously providing asymptotic stability and gait feasibility constraints satisfaction in legged systems (bipedal and multi-pedal) have been extensively addressed [1]–[3]. For instance, the works by [4] have provided rigorous model-based approaches to assign attributes such as efficiency of locomotion in off-line fashions.

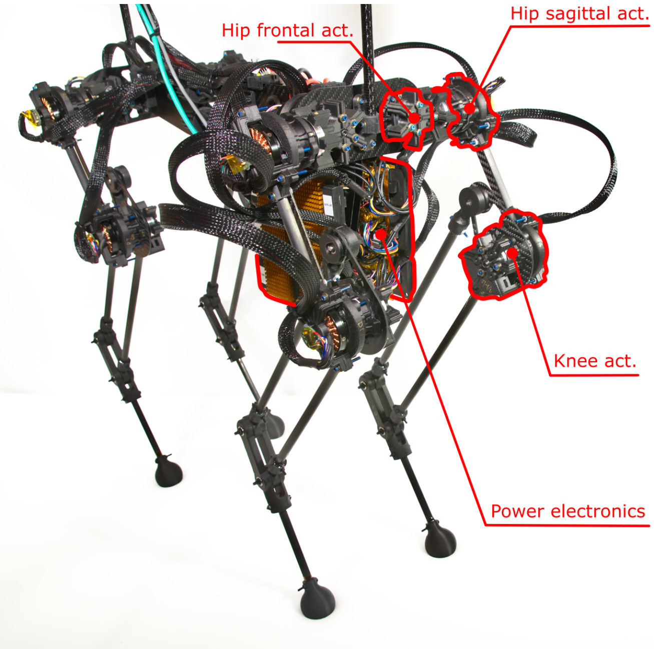

Fig. 1. NU's Husky Carbon, the platform to integrate legged and aerial mobility.

Other attempts entail optimization-based, nonlinear approaches to secure safety and performance of legged locomotion [5]–[9]. In general, in these paradigms, an optimization-based controller adjusts the gait parameters throughout the whole gait cycle such that not only the robot's posture is adjusted to accommodate the unplanned posture adjustments but also the joints position, velocity and acceleration are modified to avoid slipping into infeasible scenarios, e.g., the violation of contact forces. What makes these methods cumbersome is that they are widely defined based on whole body control at every time-step which can lead to computationally expensive optimization problems.

Many of today's real-time approaches have achieved better efficiencies through reduced-order models and decomposition methods [10], however, they still depend on optimization solvers [11]–[20]. While many optimization-based methods do not present promising horizons in experimental legged robotics, other Gauss-Newton Hessian approximation such as Sequential Linear Quadratic (SLQ) [21], iLQR and iLQG [22] have been more effective. That said, they also present

[1] ECE Department, Northeastern University, Boston, MA, USA. emails: {e.sihite, yang.yich, a.ramezani}@northeastern.edu

formidable challenges for smaller processors.

Other popular paradigms such as Approximate Dynamic Programming (ADP) [23], Reinforcement Learning (RL) [24], decoupled approaches to design control for nonlinear stochastic systems [25] can potentially remedy the challenges associated with costs of computation. However, these approaches are far from providing any practical solutions to the problem in hand and they are shown to be only effective on simpler practical robots [26]–[33] and demonstrations in more complex problems such as those offered by legged robots are missing.

In this work, we will resolve gait parameters for our quadrupedal platform off-line and re-plan them in real-time during the gait cycle. We assume well-tuned supervisory controllers found in [34]–[36] and will focus on fine-tuning the joints desired trajectories to satisfy unilateral contact force constraints. To do this, we will devise intermediary filters based on Explicit Reference Governors (ERG) [37]–[39]. ERGs relied on provable Lyapunov stability properties can perform the motion planning problem in the state space in a much faster way than widely used optimization-based methods.

II. Overview of Northeastern University's *Husky Carbon* Morpho-functional Platform

Husky Carbon, shown in Fig. 1, when standing as a quadrupedal robot, is 2.5 ft (0.8 m) tall. The robot is 12 in (0.3 m) wide. It is fabricated from reinforced thermoplastic materials through additive manufacturing and weighs 9.5 lb (4.3 kg). It hosts on-board power electronics and it operates using an external power supply. The current prototype lacks exteroceptive sensors such as camera and LiDAR. The robot is constructed of two pairs of identical legs in the form of parallelogram mechanisms. Each with three degrees-of-freedom (DOFs), the legs are fixated to Husky's torso by a one-DOF revolute joint with a large range of motion. As a result, the legs can be located sideways. This configuration allows facing the knee actuators upwards for propulsion purpose. A clutch mechanism will disengage the knee actuator from the lower limb before the actuator runs a propeller.

III. System dynamic modeling and controller

Husky can be modeled using the simple reduced order model where the leg is assumed to be massless and all masses are incorporated into the body. Then the system has 6 dynamical DOFs representing the body linear position and orientation. Each leg is modeled using two hip angles (frontal and sagittal) and a prismatic joint to describe the leg end position. This results in a very simplified model which we will use in a numerical simulation.

A. Husky Reduced Order Model (HROM)

The Husky Reduced Order Model (HROM) can be derived using Euler-Lagrangian dynamics formulation where the full conformation of Husky shown in Fig. 1 are simplified by assuming massless legs and modeling them as a variable length linkage from the hip to the foot end position. This virtual leg can defined using two rotations and length variables, and the ground reaction forces (GRF) are applied at the foot end positions which will perturb the system.

Let the superscript B represents a vector defined in the body frame (e.g. $\boldsymbol{x}^B$), and the rotation matrix R_B represents the rotation of a vector from the body frame to the inertial frame (e.g. $\boldsymbol{x} = R_B \boldsymbol{x}^B$). The foot end positions can be derived using the following kinematics equations:

$$\begin{aligned} \boldsymbol{p}_{Fi} &= \boldsymbol{p}_B + R_B \boldsymbol{l}_{hi}^B + R_B \boldsymbol{l}_{fi}^B \\ \boldsymbol{l}_{fi}^B &= R_y\left(\phi_i\right) R_x\left(\gamma_i\right) \begin{bmatrix} 0, & 0, & -r_i \end{bmatrix}^\top, \end{aligned} \tag{1}$$

where $\boldsymbol{p}_B$ is the body position, $\boldsymbol{p}_{Fi}$ is position of foot $i \in \mathcal{F}$ where $\mathcal{F}$ is the set of legs, ϕ_i and γ_i are the hip frontal and sagittal angles respectively, and r_i is the prismatic joint length.

Let $\boldsymbol{\omega}_B$ be the body angular velocity in body frame and $\boldsymbol{g}$ be the gravitational acceleration vector. The legs of HROM are massless, so we can ignore all leg states and directly calculate the kinetic energy $K = m_B \dot{\boldsymbol{p}}_B^\top \dot{\boldsymbol{p}}_B/2 + \boldsymbol{\omega}_B^\top \hat{I}_B \boldsymbol{\omega}_B/2$ and potential energy $V = -m_B \boldsymbol{p}_B^\top \boldsymbol{g}$. Then the Lagrangian of the system can be calculated as $L = K - V$, and the dynamic equation of motion can be derived using the Euler-Lagrangian Method. The body orientation is defined using the Hamiltonian's principles and the modified Lagrangian for rotation in SO(3) to avoid using Euler rotations which can become singular during the simulation. Then the equation of motion can be derived as follows:

$$\begin{aligned} &\tfrac{d}{dt}\left(\tfrac{\partial L}{\partial \dot{\boldsymbol{p}}_B}\right) - \tfrac{\partial L}{\partial \boldsymbol{p}_B} = \boldsymbol{u}_1, \qquad \dot{R}_B = R_B \left[\boldsymbol{\omega}_B\right]_\times \\ &\tfrac{d}{dt}\left(\tfrac{\partial L}{\partial \boldsymbol{\omega}_B^B}\right) + \boldsymbol{\omega}_B \times \tfrac{\partial L}{\partial \boldsymbol{\omega}_B} + \textstyle\sum_{j=1}^3 \boldsymbol{r}_{Bj} \times \tfrac{\partial L}{\partial \boldsymbol{r}_{Bj}} = \boldsymbol{u}_2, \end{aligned} \tag{2}$$

where $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are the generalized vectors, $[\,\cdot\,]_\times$ is the skew operator, and $R_B^\top = [\boldsymbol{r}_{B1}, \boldsymbol{r}_{B2}, \boldsymbol{r}_{B3}]$. The dynamic system acceleration can then be solved from into the following standard form:

$$M \begin{bmatrix} \ddot{\boldsymbol{p}}_B \\ \dot{\boldsymbol{\omega}}_B \end{bmatrix} + \boldsymbol{h} = \textstyle\sum_{i=1}^4 B_{gi}\, \boldsymbol{u}_{gi}, \tag{3}$$

where M is the mass/inertia matrix, $\boldsymbol{h}$ contains the coriolis and gravitational vectors, $B_{gi}\boldsymbol{u}_{gi}$ represent the generalized force due to the GRF ($\boldsymbol{u}_{gi}$) acting on the foot i, and $B_{gi} = (\partial \dot{\boldsymbol{p}}_{Fi} / \partial \boldsymbol{v})$ where $\boldsymbol{v} = [\dot{\boldsymbol{p}}_B^\top, \boldsymbol{\omega}_B^\top]^\top$. M, $\boldsymbol{h}$, and B_{gi} are also a function of the leg joint variables (ϕ_i, γ_i, and r_i), which is driven by setting their accelerations to track a desired joint states. The joint states and the inputs are defined as follows

$$\boldsymbol{q}_L = [\boldsymbol{\phi}^\top, \boldsymbol{\gamma}^\top, \boldsymbol{r}^\top]^\top, \qquad \ddot{\boldsymbol{q}}_L = \boldsymbol{u}_L, \tag{4}$$

where $\boldsymbol{\phi}$, $\boldsymbol{\gamma}$, and $\boldsymbol{r}$ contain the joint variables of all legs (hip frontal, hip sagittal, and leg length respectively), and $\boldsymbol{u}_L$ forms the control input to the system in the form of the leg joint state accelerations. Combining both the dynamic and massless leg states form the full system states:

$$\boldsymbol{x} = [\boldsymbol{q}_L^\top, \dot{\boldsymbol{q}}_L^\top, \boldsymbol{r}_B^\top, \boldsymbol{p}_B^\top, \dot{\boldsymbol{p}}_B^\top, \boldsymbol{\omega}_B^\top]^\top, \tag{5}$$

where the $\boldsymbol{r}_B$ contains the elements of the rotation matrix R_B. The equation of motion can then simply defined as

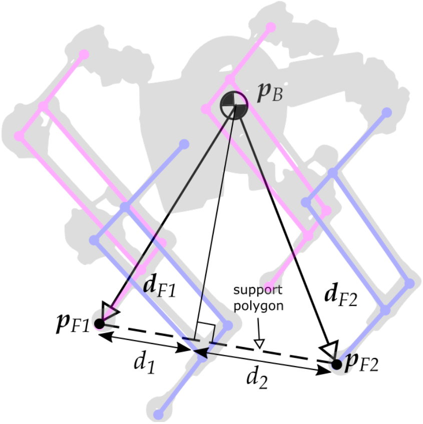

Fig. 2. The reduced order model represented by a triangular pendulum which is used for estimating GRF and constraints equations for ERG.

$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}_L, \boldsymbol{u}_g)$ which will be implemented in a numerical simulation.

The GRF is modeled using a compliant ground model and Stribeck friction model, defined as follows:

$$\begin{aligned} \boldsymbol{u}_{gi} &= \begin{cases} 0 & \text{if } z_i > 0 \\ [u_{gi,x},\, u_{gi,y},\, u_{gi,z}]^\top & \text{else} \end{cases} \\ u_{gi,z} &= -k_{gz} z_i - k_{dz} \dot{z}_i \\ u_{gi,x} &= -s_{i,x} u_{gi,z}\, \mathrm{sgn}(\dot{x}_i) - \mu_v \dot{x}_i \\ s_{i,x} &= \left(\mu_c - (\mu_c - \mu_s)\exp\left(-|\dot{x}_i|^2/v_s^2\right)\right), \end{aligned} \tag{6}$$

where x_i and z_i are the x and z positions of foot i respectively, k_{gz} and k_{dz} are the spring and damping coefficients of the compliant surface model respectively, $u_{gi,x}$ and $u_{gi,y}$ are the ground friction forces respectively, μ_c, μ_s, and μ_v are the Coulomb, static, and viscous friction coefficients respectively, and v_s is the Stribeck velocity. We omit the derivations of $u_{g,y}$ as it follows similar derivations to $u_{g,x}$.

B. Controller and Explicit Reference Governor (ERG)

The ERG framework is used to enforce the ground force constraint during the implementation of the walking gait. The ERG is utilized by defining a cost or constraint equation as a function of the controller reference. However, calculating the GRF is very complex and difficult even in HROM, so a further simplified model is used to estimate the GRF and to derive the constraint equations.

The ROM is modeled as a triangular inverted pendulum (TIP), shown in Fig. 2, which is an extension of the more standard variable length inverted pendulum model. The two contact points at the feet allows us to estimate the distribution of GRF between the two legs so we can evaluate the friction constraint on each leg individually. The kinematic constraints of the TIP model can be defined as follows:

$$\ddot{\boldsymbol{p}}_{F1} = \ddot{\boldsymbol{d}}_{F1} + \ddot{\boldsymbol{p}}_B = 0, \qquad \ddot{\boldsymbol{p}}_{F2} = \ddot{\boldsymbol{d}}_{F2} + \ddot{\boldsymbol{p}}_B = 0, \tag{7}$$

where $\boldsymbol{p}_{F1}$ and $\boldsymbol{p}_{F2}$ are the front and rear foot position respectively and $\boldsymbol{d}_{Fi}$ represent the leg position from the body CoM to either foot. Assuming no slippage at the foot end points ($\ddot{\boldsymbol{p}}_{F1} = \ddot{\boldsymbol{p}}_{F2} = 0$), the body can be controlled using the leg length acceleration $\ddot{\boldsymbol{d}}_{F1} = \ddot{\boldsymbol{d}}_{F2} = -\ddot{\boldsymbol{p}}_B$. In this case, the body acceleration $\ddot{\boldsymbol{p}}_B$ is the control input to this simplified model which will be mapped back into the joint space accelerations ($\boldsymbol{u}_L$) for HROM. Consider the following PD controller to track a desired body states ($\boldsymbol{p}_{B,ref}$) and velocity:

$$\ddot{\boldsymbol{p}}_B = K_p \left(\boldsymbol{p}_{B,ref} - \boldsymbol{p}_B\right) + K_d \left(\dot{\boldsymbol{p}}_{B,ref} - \dot{\boldsymbol{p}}_B\right), \tag{8}$$

where K_p and K_d are the PD gains for this controller. Define the vectors $\boldsymbol{x}_p = [\boldsymbol{p}_B^\top, \dot{\boldsymbol{p}}_B^\top]^\top$ and $\boldsymbol{x}_r = [\boldsymbol{p}_{B,ref}^\top, \dot{\boldsymbol{p}}_{B,ref}^\top]^\top$ which turns (8) into the concise form of $\ddot{\boldsymbol{p}}_B = K(\boldsymbol{x}_r - \boldsymbol{x}_p)$ where $K = [K_P, K_d]$.

The ground forces can be derived from the body acceleration, as this point mass dynamics can be modeled as follows:

$$m_B\, \ddot{\boldsymbol{p}}_B = m_B\, \boldsymbol{g} + \boldsymbol{u}_{g1} + \boldsymbol{u}_{g2}, \tag{9}$$

where $\boldsymbol{u}_{g1}$ and $\boldsymbol{u}_{g2}$ are the GRF of the front and rear foot respectively. Since we can only solve for the sum of the ground forces from (9), we need to make some assumptions in order to solve for $\boldsymbol{u}_{g1}$ and $\boldsymbol{u}_{g2}$. This model restricts moment about the axis perpendicular to the support polygon and by assuming the lateral ground forces are distributed evenly, we can form the following constraint equations:

$$\begin{gathered} u_{g1,y} = u_{g2,y}, \qquad |d_1|\, u_{g1,z} = |d_1|\, u_{g2,z} \\ |d_1|\, \hat{\boldsymbol{e}}_g^\top\, \boldsymbol{u}_{g1} = |d_2|\, \hat{\boldsymbol{e}}_g^\top\, \boldsymbol{u}_{g2}, \end{gathered} \tag{10}$$

where $\hat{\boldsymbol{e}}_g$ is the unit vector perpendicular to the support polygon along the ground plane, and d_1 and d_2 are the distance along the support polygon as illustrated in Fig. 2. Combining (9) and (10) yields the general form equation $A[\boldsymbol{u}_{g1}^\top, \boldsymbol{u}_{g2}^\top]^\top = \boldsymbol{b}$ where the GRF can be solved given that A is invertible. The GRF can be defined as a function of the body position reference by plugging in (8) into (9) which will be used to derive the constraint equations for the ERG.

IV. Explicit Reference Governor

Explicit reference governor (ERG) works as an add-on scheme on a pre-stabilized closed loop system. The ERG manipulates the applied reference to the controller to enforce the desired constraints while being as close as possible to the desired reference trajectory.

A. Constraint Equation Formulations

Some constraint equations are defined to enforce the no-slip condition on both stance feet of the robot. Let the constrain equation be defined as follows:

$$\boldsymbol{h}_r(\boldsymbol{x}_p, \boldsymbol{x}_r) = J_r(\boldsymbol{x}_p)\, \boldsymbol{x}_r + \boldsymbol{d}_r(\boldsymbol{x}_p) \geq 0, \tag{11}$$

where the relationship between the constraint and $\boldsymbol{x}_r$ is linear. The GRF for both feet can be solved using (8) to (10), as follows:

$$\begin{bmatrix} \boldsymbol{u}_{g1} \\ \boldsymbol{u}_{g2} \end{bmatrix} = A^{-1} \begin{bmatrix} m_B\, K\left(\boldsymbol{x}_p - \boldsymbol{x}_r\right) - m_B\, \boldsymbol{g} \\ \boldsymbol{0}_{3\times1} \end{bmatrix}, \tag{12}$$

which is linear in $\boldsymbol{x}_r$. The following GRF constraints are used to prevent slipping:

$$\begin{gathered} |u_{gi,x}| \leq \mu_s\, u_{gi,z}, \qquad |u_{gi,y}| \leq \mu_s\, u_{gi,z}, \\ u_{gi,z} \geq u_{gi,z}^{min}. \end{gathered} \tag{13}$$

Then the constraint equations can be formulated as follows:

$$\begin{bmatrix} -\mathrm{sgn}(u_{gi,x}) & 0 & \mu_s \\ 0 & -\mathrm{sgn}(u_{gi,y}) & \mu_s \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{u}_{gi} + \begin{bmatrix} 0 \\ 0 \\ -u_{gi,z}^{min} \end{bmatrix} \geq 0, \tag{14}$$

for leg $i \in 1, 2$. The J_r and $\boldsymbol{d}_r$ in (11) can be calculated numerically using (12) and (14) which will be used to formulate the ERG applied reference update law.

B. ERG Formulations

Algorithm 1 ERG algorithm to constraint GRFs

```
$\boldsymbol{h}_r = \boldsymbol{h}_r(\boldsymbol{x}_p, \boldsymbol{x}_r)$
$\boldsymbol{h}_w = \boldsymbol{h}_r(\boldsymbol{x}_p, \boldsymbol{x}_w)$
$\boldsymbol{v}_r = \boldsymbol{v}_t = \boldsymbol{v}_n = \boldsymbol{0}$
$C_r = []$
if $\min(\boldsymbol{h}_w) \geq 0$ or $\min(\boldsymbol{h}_r) \geq 0$ then
    $v_r = \alpha_r (x_r - x_w)$
end if
if $\min(\boldsymbol{h}_w) \geq 0$ and $\min(\boldsymbol{h}_r) < 0$ then
    $n_c = \mathrm{length}(\boldsymbol{h}_r)$
    for $k = 1 : n_c$ do
        $v_r = \alpha_r (x_r - x_w)$
        if $h_{r,i} < 0$ then
            $C_r = [\boldsymbol{C}_r; \boldsymbol{J}_r(k, :)]$
        end if
    end for
    $\boldsymbol{N}_r = \mathrm{null}(\boldsymbol{C}_r)$
    $[\sim, n] = \mathrm{size}(\boldsymbol{N}_r)$
    $\boldsymbol{v}_t = \boldsymbol{0}$
    for $k = 1 : n$ do
        $\boldsymbol{n}_k = \boldsymbol{N}_r(:, k)/\boldsymbol{N}_r(:, k)$
        $\boldsymbol{v}_t = \boldsymbol{v}_t + \alpha_t \boldsymbol{n}_k \boldsymbol{n}_k^\top (\boldsymbol{x}_r - \boldsymbol{x}_w)$
    end for
end if
if $\min(\boldsymbol{h}_w) < 0$ and $\min(\boldsymbol{h}_r) < 0$ then
    $k_{min} = \arg\min_k(\boldsymbol{h}_w)$    ▷ index of the smallest $h_w$
    $\boldsymbol{r}_k = \boldsymbol{J}_r(k_{min}, :)/\boldsymbol{J}_r(k_{min}, :)$
    if $h_r(k_{min}) \geq h_w(k_{min})$ then
        $\boldsymbol{v}_n = \alpha_n \boldsymbol{r}_k \boldsymbol{r}_k^\top (\boldsymbol{x}_r - \boldsymbol{x}_w)$
    else
        $\boldsymbol{v}_n = -\alpha_n \boldsymbol{r}_k \boldsymbol{r}_k^\top (\boldsymbol{x}_r - \boldsymbol{x}_w)$
    end if
end if
$\dot{\boldsymbol{x}}_w = \boldsymbol{v}_r + \boldsymbol{v}_t + \boldsymbol{v}_n$
$\boldsymbol{x}_w = \boldsymbol{x}_w + \Delta t \dot{\boldsymbol{x}}_w$
```

The ERG framework is utilized to enforce the friction pyramid constraint in (14) which is formulated into the form shown in (11). The ERG algorithm manipulates the applied reference into the controller in (8) such that (11) is satisfied.

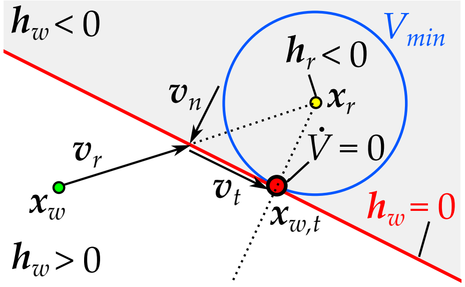

Fig. 3. Explicit Reference Governor (ERG) manipulates the applied reference states $\boldsymbol{x}_w$ to be as close as possible to the desired reference $\boldsymbol{x}_r$ without violating the constraint equation $\boldsymbol{h}_w \geq 0$.

This method is very useful as it avoids using optimization or nonlinear MPC frameworks to enforce constraints on the harder-to-model GRFs.

Let $\boldsymbol{x}_w$ be the applied reference which will be used in the controller instead of $\boldsymbol{x}_r$, and $\boldsymbol{h}_w = \boldsymbol{h}_r(\boldsymbol{x}_p, \boldsymbol{x}_w)$ be the constraint equation value using $\boldsymbol{x}_w$. The ERG manipulates the applied reference ($\boldsymbol{x}_w$) to avoid violating the constraint equation $\boldsymbol{h}_w \geq 0$ while also be as close as possible to the desired reference ($\boldsymbol{x}_r$), as illustrated in Fig. 3. Consider the Lyapunov equation $V = (\boldsymbol{x}_r - \boldsymbol{x}_w)^\top P(\boldsymbol{x}_r - \boldsymbol{x}_w)$. $\boldsymbol{x}_w$ is updated through the update law:

$$\dot{\boldsymbol{x}}_w = \boldsymbol{v}_r + \boldsymbol{v}_t + \boldsymbol{v}_n, \tag{15}$$

where $\boldsymbol{v}_r$ drives $\boldsymbol{x}_w$ directly to $\boldsymbol{x}_r$, while $\boldsymbol{v}_t$ and $\boldsymbol{v}_n$ drives $\boldsymbol{x}_w$ along the surface and into the boundary $\boldsymbol{h}_w = 0$, respectively. The objective of this ERG algorithm is to drive $\boldsymbol{x}_w$ to the state $\boldsymbol{x}_{w,t}$ which is the minimum energy solution V_{min} that satisfies the constraint $\boldsymbol{h}_w \geq 0$. The overall structure of this algorithm can be seen in Algorithm 1.

Let C_r be the rowspace of the violated constraints of $\boldsymbol{h}_r$ (rows of the J_r where the constraint is violated). Define $N_r = \mathrm{null}(C_r) = [\boldsymbol{n}_1, \ldots, \boldsymbol{n}_n]$ where n is the size of the nullspace. Additionally, let $\boldsymbol{r}_k$ be the k'th row of J_r. Then the following update law is used for $\boldsymbol{v}_r$, $\boldsymbol{v}_t$, and $\boldsymbol{v}_n$:

$$\begin{gathered} \boldsymbol{v}_r = \hat{\alpha}_r (\boldsymbol{x}_r - \boldsymbol{x}_w), \qquad \boldsymbol{v}_n = \hat{\alpha}_n\, \boldsymbol{r}_k\, \boldsymbol{r}_k^\top (\boldsymbol{x}_r - \boldsymbol{x}_w) \\ \boldsymbol{v}_t = \textstyle\sum_{k=1}^{n} \hat{\alpha}_t\, \boldsymbol{n}_k\, \boldsymbol{n}_k^\top (\boldsymbol{x}_r - \boldsymbol{x}_w), \end{gathered} \tag{16}$$

where $\hat{\alpha}$ are scalars defined as follows:

$$\begin{aligned} \hat{\alpha}_r &= \begin{cases} \alpha_r, & \text{if } \min(\boldsymbol{h}_w) \geq 0 \text{ or } \min(\boldsymbol{h}_r) \geq 0 \\ 0, & \text{else} \end{cases} \\ \hat{\alpha}_t &= \begin{cases} \alpha_t, & \text{if } \min(\boldsymbol{h}_w) \geq 0 \text{ or } \min(\boldsymbol{h}_r) < 0 \\ 0, & \text{else} \end{cases} \\ \hat{\alpha}_n &= \begin{cases} \alpha_n, & \text{if } \min(\boldsymbol{h}_w) \leq \min(\boldsymbol{h}_r) < 0 \\ -\alpha_n, & \text{if } \min(\boldsymbol{h}_r) < \min(\boldsymbol{h}_w) < 0 \\ 0, & \text{else,} \end{cases} \end{aligned} \tag{17}$$

where α is a positive scalar which determines the rate of convergence.

Assuming $\dot{\boldsymbol{x}}_r = 0$ and using the update law defined from (16) and (17), we will have $\dot{V} = -2(\boldsymbol{x}_r - \boldsymbol{x}_w)^\top Q\, (\boldsymbol{x}_r - \boldsymbol{x}_w)$, with $Q = P(\hat{\alpha}_r I + \sum_{k=1}^{n} \hat{\alpha}_t\, \boldsymbol{n}_k\, \boldsymbol{n}_k^\top + \hat{\alpha}_n\, \boldsymbol{r}_k\, \boldsymbol{r}_k^\top)$. We have

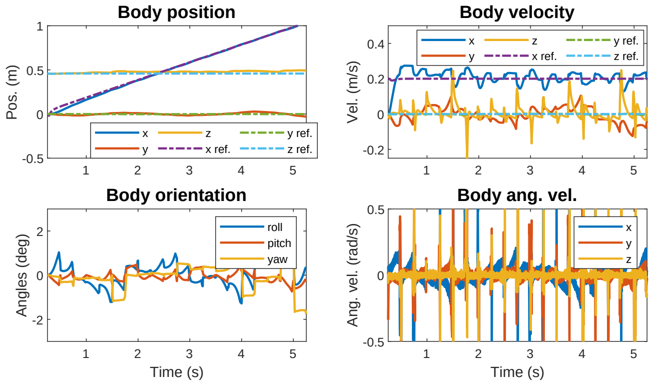

Fig. 4. Simulated states of the Husky as it walks stably on a slippery surface ($\mu_s = 0.2$) for 20 gait periods.

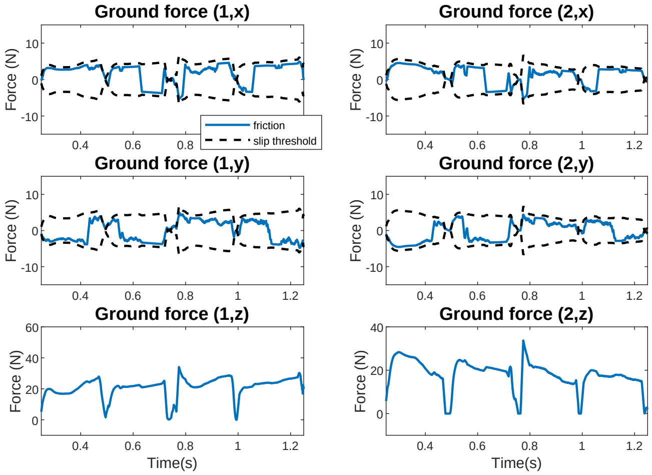

Fig. 5. The simulated GRF on the front and rear stance foot, labeled as 1 and 2 respectively, during the first four gait periods. The ground friction stayed within the friction pyramid constraints which are shown as the dashed lines.

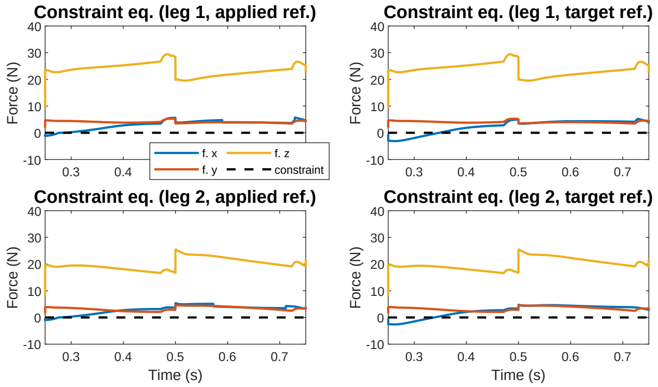

Fig. 6. The constraint equation values between using the applied vs. target reference during the first two gait periods. The initial reference cause constraint violation which is avoided using the ERG update law.

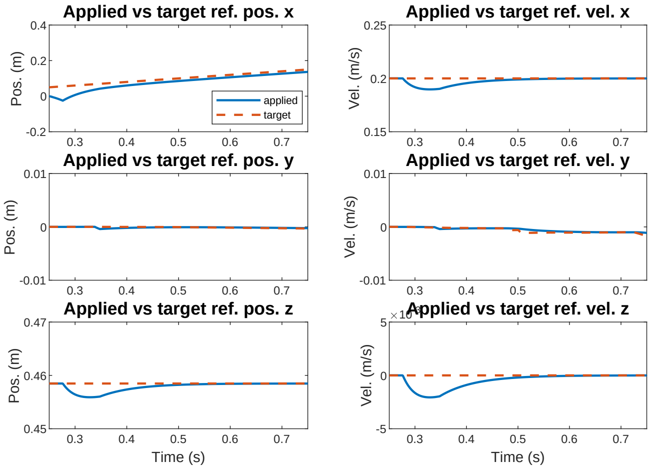

Fig. 7. The target vs applied references (body linear position and velocity) during the period of constraint violation of the target references. The applied reference is manipulated to avoid breaking the constraints.

the gradient $\dot{V} = 0$ if $\min(\boldsymbol{h}_w) \leq 0$ and $\boldsymbol{n}_k \perp (\boldsymbol{x}_r - \boldsymbol{x}_w)$, while $\dot{V} < 0$ when $\min(\boldsymbol{h}_r) \geq 0$ or when $\min(\boldsymbol{h}_w) \geq 0$. In case both applied reference and target constraints equation are violated, we have $\dot{V} > 0$ which drives the $\boldsymbol{x}_w$ towards the constraint boundary. This allows the $\boldsymbol{x}_w$ to converge to $\boldsymbol{x}_{w,t}$ which is the minimum energy solution that satisfies $\boldsymbol{h}_w \geq 0$ as illustrated in Fig. 3.

V. Numerical Simulation

The simulation is implemented on HROM based on a predefined quasi-static trotting gait in MATLAB. The stance-swing pair switching is done by using a timing-based state machine, where the gait period is defined as 0.25 s and a total of 20 steps is simulated. The input of the ERG and the stance foot controller in (8) is defined as the the inertial frame foot end acceleration relative to the current body position. To implement this controller, we simply integrated this acceleration to the desired position and velocity about the body frame, and track these positions using a PID controller. The swing foot trajectory follows a simple bezier curve trajectory with an end position located at a fixed position relative to the body (0.08 m ahead of hip position in x-axis). The bezier curve trajectory initial position is set as the previous stance foot final position and a target swing height of 0.2 m with zero initial and final velocity. Since we have no row, pitch or yaw compensation, we rotate the reference by $R_B^\top$ at every timestep so that we robot acts like walking on a flat surface without angular deviation.

The simulation results of the robot walking on slippery surface ($\mu_s = 0.2$) at the target forward speed of 0.2 m/s is shown in Fig. 4 to 7. The resulting gait is stable, with a potential slipping happening at the beginning of the gait which is regulated by the ERG as shown in Fig. 5 and 7. The ERG has successfully manipulated the controller reference states to avoid constraint violation which resulted in a stable trotting gait even on a slippery surface. Figure 5 shows the actual GRF applied to this robot during this simulation and the friction is kept within the constraint boundary defined in (13).

VI. Conclusions and future work

We explored the performance of the ERG algorithm on a quadruped where the no-slip constraints have been enforced successfully. The use of ERG allows the robot to walk stably on a slippery surface ($\mu_s = 0.2$) where the controller states are manipulated to satisfy the GRF constraints we specified. This algorithm can be computed quickly, which means that it can serve as an alternative the more computationally expensive algorithms for enforcing constraints, such as optimization and whole body control.

The controller considered in this work is only defined for the stance foot where all of the GRF are applied. Future work should include the usage of the swing foot placement algorithm to regulate the robot heading and the tracking a more complex trajectories. Additionally, other constraints can also be considered for the ERG, such as input and state saturation constraints which is practical for the implementation in the actual robot.

References

[1] T. Apgar, P. Clary, K. Green, A. Fern, and J. Hurst, "Fast Online Trajectory Optimization for the Bipedal Robot Cassie," in *Robotics: Science and Systems XIV*, 2018.

[2] G. Bledt, "Regularized predictive control framework for robust dynamic legged locomotion," Ph.D. dissertation, Massachusetts Institute of Technology, 2020.

[3] M. Hutter *et al.*, "ANYmal - a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 38–44.

[4] E. Westervelt and J. Grizzle, *Feedback Control of Dynamic Bipedal Robot Locomotion*, ser. Control and Automation Series. CRC PressINC, 2007. [Online]. Available: https://books.google.com/books?id=xaMeAQAAIAAJ

[5] Q. Nguyen and K. Sreenath, "Optimal robust control for bipedal robots through control lyapunov function based quadratic programs." in *Robotics: Science and Systems*, 2015.

[6] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, "Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs," *IEEE Access*, vol. 3, pp. 323–332, 03 2015.

[7] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly Exponentially Stabilizing Control Lyapunov Functions and Hybrid Zero Dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, Apr. 2014.

[8] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1447–1454.

[9] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 2641–2648.

[10] T. Buschmann, S. Lohmeier, M. Bachmayer, H. Ulbrich, and F. Pfeiffer, "A collocation method for real-time walking pattern generation," 2007, pp. 1–6.

[11] T. Koolen *et al.*, "Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas," *International Journal of Humanoid Robotics*, vol. 13, no. 01, 2016.

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

[13] C. Dario Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov. 2016, pp. 558–564, iSSN: 2164-0580.

[14] T. Bretl and S. Lall, "Testing Static Equilibrium for Legged Robots," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 794–807, Aug. 2008.

[15] D. Pardo, L. Möller, M. Neunert, A. W. Winkler, and J. Buchli, "Evaluating Direct Transcription and Nonlinear Optimization Methods for Robot Motion Planning," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 946–953, Jul. 2016.

[16] C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini, "Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1096–1103.

[17] C. Mastalli, I. Havoutis, A. W. Winkler, D. G. Caldwell, and C. Semini, "On-line and on-board planning and perception for quadrupedal locomotion," in *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, May 2015, pp. 1–7.

[18] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and Trajectory Optimization for Legged Systems Through Phase-Based End-Effector Parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, Jul. 2018.

[19] B. Aceituno-Cabezas *et al.*, "Simultaneous Contact, Gait, and Motion Planning for Robust Multilegged Locomotion via Mixed-Integer Convex Optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2531–2538, Jul. 2018.

[20] H. Dai and R. Tedrake, "Planning robust walking motion on uneven terrain via convex optimization," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov. 2016, pp. 579–586, iSSN: 2164-0580.

[21] A. Sideris and J. E. Bobrow, "An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems," in *American Control Conference*. IEEE, 2005, pp. 2275–2280.

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

[23] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007, vol. 703.

[24] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[25] M. Rafieisakhaei, S. Chakravorty, and P. Kumar, "A near-optimal separation principle for nonlinear stochastic systems arising in robotic path planning and control," *arXiv preprint arXiv:1705.08566*, 2017.

[26] G. Berseth, C. Xie, P. Cernek, and M. Van de Panne, "Progressive Reinforcement Learning with Distillation for Multi-Skilled Motion Control," *arXiv:1802.04765 [cs, stat]*, Feb. 2018, arXiv: 1802.04765. [Online]. Available: http://arxiv.org/abs/1802.04765

[27] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. A. Eslami, M. Riedmiller, and D. Silver, "Emergence of Locomotion Behaviours in Rich Environments," *arXiv:1707.02286 [cs]*, Jul. 2017, arXiv: 1707.02286. [Online]. Available: http://arxiv.org/abs/1707.02286

[28] X. Peng, B. Sun, K. Ali, and K. Saenko, "Learning deep object detectors from 3d models," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1278–1286.

[29] Z. Xie, G. Berseth, P. Clary, J. Hurst, and M. van de Panne, "Feedback control for cassie with deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1241–1246.

[30] T. Haarnoja, V. Pong, A. Zhou, M. Dalal, P. Abbeel, and S. Levine, "Composable deep reinforcement learning for robotic manipulation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6244–6251.

[31] A. R. Mahmood, D. Korenkevych, G. Vasan, W. Ma, and J. Bergstra, "Benchmarking Reinforcement Learning Algorithms on Real-World Robots," *arXiv:1809.07731 [cs, stat]*, Sep. 2018. [Online]. Available: http://arxiv.org/abs/1809.07731

[32] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, "Learning to Walk via Deep Reinforcement Learning," *arXiv:1812.11103 [cs, stat]*, Jun. 2019, arXiv: 1812.11103. [Online]. Available: http://arxiv.org/abs/1812.11103

[33] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. Sydney, NSW, Australia: JMLR.org, Aug. 2017, pp. 1352–1361.

[34] E. D. Sontag, "A lyapunov-like characterization of asymptotic controllability," *SIAM journal on control and optimization*, vol. 21, no. 3, pp. 462–471, 1983.

[35] P. V. Kokotovic, M. Krstic, and I. Kanellakopoulos, "Backstepping to passivity: recursive design of adaptive systems," in *IEEE Conference on Decision and Control*, vol. 4, 12 1992, pp. 3276–3280.

[36] S. P. Bhat and D. S. Bernstein, "Continuous finite-time stabilization of the translational and rotational double integrators," *IEEE Transactions on Automatic Control*, vol. 43, no. 5, pp. 678–682, 05 1998.

[37] E. G. Gilbert, I. Kolmanovsky, and Kok Tin Tan, "Nonlinear control of discrete-time linear systems with state and control constraints: a reference governor with global convergence properties," in *IEEE Conference on Decision and Control*, vol. 1, 12 1994, pp. 144–149.

[38] A. Bemporad, "Reference governor for constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 3, pp. 415–419, 1998.

[39] E. Gilbert and I. Kolmanovsky, "Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor," *Automatica*, vol. 38, no. 12, pp. 2063–2073, 2002.