Proceedings of the ASME 2014 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2014
August 17-20, 2014, Buffalo, New York, USA

DETC2014-34374

AN EVALUATION OF MOREAU'S TIME-STEPPING SCHEME FOR THE SIMULATION OF A LEGGED ROBOT

Christian Gehring*
Remo Diethelm
Roland Siegwart
Autonomous Systems Lab, ETH Zurich
8092 Zurich, Switzerland

Gabriel Nützi
Remco I. Leine
Center of Mechanics, ETH Zurich
8092 Zurich, Switzerland

ABSTRACT

A state-of-the-art simulation technique that solves the equations of motion together with the set-valued contact and impulse laws by the time-stepping scheme of Moreau is introduced to the legged robotics community. An analysis is given that shows which of the many variations of the method fits best to legged robots. Two different methods to solve the discretized normal cone inclusions are compared: the projected over-relaxed Jacobi and Gauss-Seidel iteration. The methods are evaluated for an electrically-driven quadrupedal robot in terms of robustness, accuracy, speed and ease of use. Furthermore, the dependence of the simulation speed on the choice of the generalized coordinates is examined. The proposed technique is implemented in C++ and compared to a fast and simple approach based on compliant contact models. In conclusion, the introduced method with hard contacts is very beneficial for the simulation of legged robots.

INTRODUCTION

Research on legged robots heavily relies on multibody simulations, which are required to safely test new locomotion control algorithms or may be used for motion planning and design optimizations. Simulators for such robots need to cope with high-dimensional systems and non-smooth dynamics due to unilateral frictional contacts and impacts. At the same time they should be efficient for real-time control algorithms and very accurate to avoid any unrealistic opportunities that an optimization could possibly exploit. Most existing simulation tools used in robotics cannot meet all of these demanding requirements. Therefore, the development of powerful physics engines for legged robotics is a topic of active interest, in particular, as legged machines become more and more complex.

The speed requirement is most often a knock-out criterion in robotics and therefore efficient computation methods for the forward dynamics have been developed, for instance, the Articulated-Rigid-Body algorithm or the Composite-Rigid-Body method [1]. When contacts are involved, compliant contact models are mostly preferred to increase simulation speed, since their evaluation is considered to be very cheap compared to hard contact models which need numerical iterations. This may however be a fallacy, because compliant contact models may require more simulation steps to generate accurate and robust solutions than others. To tackle the numerical issues, different compliant models, even tailored for legged robots, have been proposed and analyzed [2, 3]. The major shortcomings of such simple models however remain: their parameters have to be tuned manually, which is a tedious process as they often have both numerical and physical effects.

There exists a variety of general-purpose physics engines [4, 5], often designed for games, but widely used in robotics [6, 7]. Such physics engines are able to simulate intricate virtual reality environments, but often make use of approximations of the constraints, which may result in unrealistic motions. Commonly,

[*] Address all correspondence to this author: gehrinch@ethz.ch

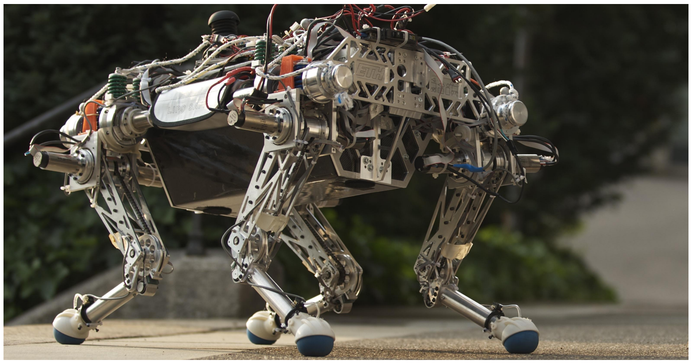


FIGURE 1: QUADRUPED ROBOT STARLETH

the mechanical system is described using a set of coordinates that define the pose of each body in Cartesian space while enforcing the joint constraints numerically and approximate the non-smooth dynamics, for instance, by formulating the contact dynamics as linear complementary problems [8]. These shortcomings that originate from a trade-off between speed, universal applicability and verisimilitude are not always acceptable for the simulation of legged robotics and may be far from optimal.

Various simulators were developed for legged robots in recent years. Kanehiro et al. [9] developed their own simulation technique to simulate a humanoid robot. A simulator for a humanoid with compliant contacts was developed by Dallali et al. [10]. Todorov et al. [8] use a discrete-time velocity-based formulation based on the work of Stewart et al. [11], and could achieve impressive simulation speeds. Their simulator is however closed source and designed as multi-purpose engine, which may not exploit all characteristics of a legged robot.

In this paper, we introduce a state-of-the-art simulation technique to the legged robotics community, which was developed during the past years in the field of mechanics and was successfully applied to various mechanical problems [12], including a robotic snake [13]. The proposed method solves the equations of motion together with the set-valued contact laws and the impact equations by the time-stepping scheme of Moreau [14]. The set-valued contact and impulse laws are modeled as discretized normal cone inclusions, which need to be solved iteratively. Two popular numeric iteration methods are thus analyzed in this work: the projected over-relaxed Jacobi (JOR) and the Gauss-Seidel (SOR) iteration.

Due to the distinctive structure of legged robots (large, heavy trunk and thin, lightweight legs), a subtle choice of the coordinates to describe the multibody system may significantly improve the accuracy and speed of the simulation. Three different sets of coordinates are therefore examined in this work.

We investigate which of the many variations of the method with hard contacts fits best to legged robots and provide an evaluation of the overall performance. We focus on the simulation of a medium-dog-sized electrically-driven quadrupedal robot called StarlETH [15] shown in Fig. 1. The quadruped with 18-degrees-of-freedom (DOF) is capable of various dynamic motions including fast walking, trotting, and pronking. The locomotion control algorithms were so far developed with a simple simulation technique using compliant contacts in combination with a standard Runge-Kutta integrator [16]. The current paper thus compares the newly proposed and previously used method in terms of robustness, accuracy, speed and user-friendliness.

NON-SMOOTH DYNAMICS

We first introduce the simulation algorithm based on Moreau's time-stepping scheme together with the *hard* and *compliant* contact model to solve the equations of motion of scleronomic dynamical systems. For a more complete introduction to non-smooth dynamics briefly covered in the next sections, the reader is referred to [17].

Equations of Motion

The equations of motion of a non-smooth scleronomic dynamical system (cf. [18]) described with the generalized coordinates $\mathbf{q}$ and velocities $\mathbf{u}$ can be represented by a set of equations of the form

$$\begin{aligned} \mathbf{M}(\mathbf{q})\,\dot{\mathbf{u}} - \mathbf{h}(\mathbf{q},\mathbf{u}) - \mathbf{W}_U\boldsymbol{\lambda}_U - \mathbf{W}_B\boldsymbol{\lambda}_B &= \mathbf{0}, \\ \dot{\mathbf{q}} &= \mathbf{F}(\mathbf{q})\,\mathbf{u} \\ \boldsymbol{\gamma}_{U_i} &= \mathbf{W}_{U_i}^\top \mathbf{u} \quad \forall i \in \mathcal{I}(\mathbf{q}) \\ \mathbf{g}_{B_j} = \mathbf{0} \quad \text{and} \quad \boldsymbol{\gamma}_{B_j} &= \mathbf{W}_{B_j}^\top \mathbf{u} \quad \forall j. \end{aligned} \tag{1}$$

The symmetric and positive definite mass matrix in Eqn. (1) is denoted by $\mathbf{M}$. The non-linear terms including Coriolis terms, centripetal terms and impressed generalized forces such as gravitational forces and motor torques are denoted in Eqn. (1) by the vector $\mathbf{h}$. The Matrix $\mathbf{F}(\mathbf{q})$ in Eqn. (1) maps the generalized velocities to the derivative of the generalized coordinates as explained later in Eqn. (22). The contact forces $\boldsymbol{\lambda}_U = [\ldots, \boldsymbol{\lambda}_{U_i}^\top, \ldots]^\top$ and the generalized force directions $\mathbf{W}_U(\mathbf{q}) = [\ldots, \mathbf{W}_{U_i}(\mathbf{q}), \ldots]$ in Eqn. (1) model the interaction between the rigid bodies, which is described by a set of closed contacts. A contact i between two interacting bodies is said to be active if it is closed on displacement level. This gives the definition for the set $\mathcal{I}(\mathbf{q})$ in Eqn. (1) as $\mathcal{I}(\mathbf{q}) := \{i \mid g_{N_i}(\mathbf{q}) = 0\}$ where $g_{N_i}(\mathbf{q})$ is the gap function in normal direction of a contact i as illustrated in Fig. 2 with the contact points P and Q. The vector $\boldsymbol{\gamma}_{U_i}(\mathbf{q})$ in Eqn. (1) denotes the relative velocity of a closed contact i which is a linear function of $\mathbf{u}$. The relative velocity $\boldsymbol{\gamma}_{U_i}$ of contact i is linked to the corresponding generalized force $\boldsymbol{\lambda}_{U_i}$ by set-valued force laws for the case of modeling the contact as a *hard contact*, and by explicit smooth force laws in

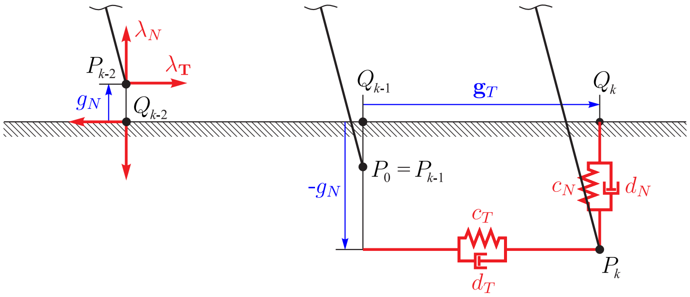


FIGURE 2: CONTACT STATE AND COMPLIANT MODEL

the case of modeling it as a *compliant contact*. The generalized bilateral forces $\boldsymbol{\lambda}_B = [\ldots, \boldsymbol{\lambda}_{B_j}^\top, \ldots]^\top$ together with their relative velocities $\frac{\mathrm{d}}{\mathrm{dt}}\mathbf{g}_{B_j} = \boldsymbol{\gamma}_{B_j}\ \forall j$ and their generalized directions $\mathbf{W}_B(\mathbf{q}) = [\ldots, \mathbf{W}_{B_j}(\mathbf{q}), \ldots]$ are used to incorporate all bilateral constraints j defined on the mechanical system, which are treated as set-valued force laws as well. In this work, the *hard contact* model encompasses two set-valued force laws, namely the unilateral contact and the spatial Coulomb friction. The force laws of the hard and compliant contact models are briefly explained in the next sections. More information about the constitutive set-valued force laws can be found in [18, 19] and about the compliant contact model in [16].

Unilateral Contact

During the contact of two bodies, a normal force λ_N acts on each body at the contact point. The associated set-valued force law for the *hard contact* model for this (closed) contact can be expressed on velocity level as

$$g_N = 0: \qquad \gamma_N \in \mathcal{N}_{\mathcal{C}_N}(-\lambda_N), \quad \mathcal{C}_N = \mathbb{R}_0^-. \tag{2}$$

If the gap between two bodies in normal direction g_N is open ($g_N > 0$) then the two bodies do not share any contact point. If the normal gap is zero and the normal relative velocity γ_N is zero as well, the normal force λ_N is positive or zero. Conversely, if $\gamma_N > 0$ then λ_N is zero because the unilateral contact is opening. This behavior is reproduced by the normal cone inclusion in Eqn. (2) to the convex set $\mathbb{R}_0^-$, which is visualized in Fig. 3a by the thick lines.

The discontinuous *compliant contact* model used in this work approximates the unilateral contact by spring-damper elements between the current (P) and the first detected (P_0) contact point as indicated in Fig. 2. When the contact is closed, the normal component of the contact force λ_N is computed using the spring stiffness c_N and damping coefficient d_N as shown in Alg. 1. Note that λ_N is strictly non-negative which is taken into account on line 1 in Alg. 1.

ALGORITHM 1 COMPLIANT CONTACT LAW IF CLOSED

```
 1: $\lambda_N \leftarrow \max(-c_N\, g_N - d_N\, \gamma_N, 0)$
 2: if $Slip == true$ then
 3:     if $\|\gamma_T\|_2 \approx 0$ or $\|\boldsymbol{\lambda}_T\|_2 \leq \mu\lambda_N$ then
 4:         $Slip \leftarrow false$,   Reset point $P_0$
 5:     end if
 6: else
 7:     $\boldsymbol{\lambda}_T \leftarrow -c_T\, \mathbf{g}_T - d_T\, \boldsymbol{\gamma}_T$
 8:     if $\|\boldsymbol{\lambda}_T\|_2 \geq \mu\lambda_N$ then
 9:         $\boldsymbol{\lambda}_T \leftarrow -\mu\lambda_N \frac{\boldsymbol{\gamma}_T}{\|\boldsymbol{\gamma}_T\|_2}$,   $Slip \leftarrow true$
10:     end if
11: end if
```

Spatial Coulomb Friction

Spatial Coulomb friction can be used to model a frictional contact between two bodies. The friction force $\boldsymbol{\lambda}_T := [\lambda_{T_1}, \lambda_{T_2}]^\top$ acts in the tangential plane between the two bodies. In case of slipping, the friction force $\boldsymbol{\lambda}_T$ points in the opposite direction of the relative velocity $\boldsymbol{\gamma}_T := [\gamma_{T_1}, \gamma_{T_2}]^\top$, and in case of sticking its upper magnitude is proportional to the normal contact force λ_N by the friction parameter μ. The spatial Coulomb friction is a set-valued force law which is used for the *hard contact* model and can be expressed as a normal cone inclusion on velocity level as

$$\boldsymbol{\gamma}_T \in \mathcal{N}_{\mathcal{C}_T}(-\boldsymbol{\lambda}_T), \quad \mathcal{C}_T(\mu\lambda_N) = \mathcal{R}_2 \cdot \mu\lambda_N, \tag{3}$$

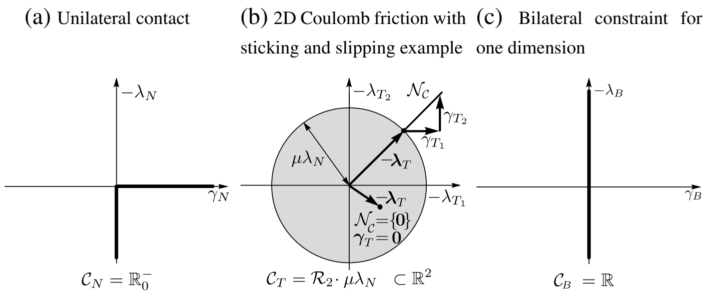
where the convex set $\mathcal{C}_T$ is the friction disc in $\mathbb{R}^2$ with the closed unit ball denoted by $\mathcal{R}_2 := \{x \in \mathbb{R}^2 |\ \|x\|_2 \leq 1\}$ and illustrated in Fig. 3b. The normal cone in Eqn. (3) models sticking behavior (relative tangential velocity $\boldsymbol{\gamma}_T = \mathbf{0}$) if the friction force $\boldsymbol{\lambda}_T$ is in the interior of the set $\mathcal{C}_T$, and models slipping behavior ($\boldsymbol{\gamma}_T$ proportional to $-\boldsymbol{\lambda}_T$) if $\boldsymbol{\lambda}_T$ lies on the boundary of $\mathcal{C}_T$ ($\|\boldsymbol{\lambda}_T\| = \mu\lambda_N$).

Algorithm 1 shows how the *compliant contact* model approximates this friction law with spring stiffness c_T and damping coefficient d_T. As soon as the magnitude of the static friction force $\boldsymbol{\lambda}_T$ (line 8) is larger than $\mu\lambda_N$, the contact starts to glide and $\boldsymbol{\lambda}_T$ is set to the dynamic friction force (line 9). Similarly in the case of slipping, if the relative velocity $\boldsymbol{\gamma}_T$ becomes almost zero or the tangential force is smaller than $\mu\lambda_N$, then point P_0 is reset and the sticking case is applied in the next time step.

For a contact i, the contact force $\boldsymbol{\lambda}_{U_i}$ in Eqn. (1) can now be stated as $\boldsymbol{\lambda}_{U_i} := [\lambda_{N_i}, \boldsymbol{\lambda}_{T_i}^\top]^\top$, $\boldsymbol{\gamma}_{U_i} := [\gamma_{N_i}, \boldsymbol{\gamma}_{T_i}^\top]^\top \in \mathbb{R}^3$.

Bilateral Constraint

Bilateral constraints prevent disallowed displacements and/or rotations of a body relative to another. The associated set-valued force law for a d_j-dimensional constraint $\mathbf{g}_{B_j} = \mathbf{0}$ in



(a) Unilateral contact (b) 2D Coulomb friction with sticking and slipping example (c) Bilateral constraint for one dimension

$-\lambda_N$
γ_N
$\mathcal{C}_N = \mathbb{R}_0^-$
$-\lambda_{T_2}$
$\mathcal{N}_\mathcal{C}$
γ_{T_2}
γ_{T_1}
$\mu\lambda_N$
$-\boldsymbol{\lambda}_T$
$-\boldsymbol{\lambda}_T$
$-\lambda_{T_1}$
$\mathcal{N}_\mathcal{C}=\{\mathbf{0}\}$
$\boldsymbol{\gamma}_T=\mathbf{0}$
$\mathcal{C}_T = \mathcal{R}_2 \cdot \mu\lambda_N \subset \mathbb{R}^2$
$-\lambda_B$
γ_B
$\mathcal{C}_B = \mathbb{R}$

FIGURE 3: RANGES OF THE SET-VALUED FORCE LAWS

Eqn. (1) can be stated on velocity level as normal cone inclusion to the convex set $\mathcal{C}_{B_j} = \mathbb{R}^{d_j}$ (see Fig. 3c) as

$$\mathbf{0} \overset{!}{=} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{g}_{B_j}(\mathbf{q}) = \boldsymbol{\gamma}_{B_j} \in \mathcal{N}_{\mathcal{C}_{B_j}}(-\boldsymbol{\lambda}_{B_j}),\ \ \boldsymbol{\gamma}_{B_j}, \boldsymbol{\lambda}_{B_j} \in \mathbb{R}^{d_j}. \quad (4)$$

Newton-Type Impacts

To include impacts in the equations of motion in Eqn. (1) which act as Dirac contributions at certain time instants, the set-valued force laws have to be supplemented with an impact law. During the collision of two bodies and the successive separation afterwards, the relative velocity behaves discontinuous at the impact time (jumps in the velocity-time profile). The generalized coordinates will be continuous since the relative velocities stay finite. The original Newton impact law relates pre-impact relative velocity $\boldsymbol{\gamma}_k^-$ to the post-impact relative velocity $\boldsymbol{\gamma}_k^+$ by $\boldsymbol{\gamma}_k^+ = -\epsilon_k \boldsymbol{\gamma}_k^-$ where $\epsilon_k \in [0, 1]$ denotes the restitution coefficient. In this paper, a *Newton-type* impact law is used for each normal cone inclusion. For more information about impacts, the reader is referred to [19]. The associated *Newton-type* impact inclusion can be formulated with the pre-impact velocity $\boldsymbol{\gamma}_k^-$ and post-impact velocity $\boldsymbol{\gamma}_k^+$ as

$$\boldsymbol{\gamma}_k^+ + \epsilon_k \boldsymbol{\gamma}_k^- := \boldsymbol{\xi}_k \in \mathcal{N}_{\mathcal{D}_k}(-\boldsymbol{\Lambda}_k). \quad (5)$$

The impulsive force (also called percussion) $\boldsymbol{\Lambda}_k$ is the integrated Dirac contribution of the force $\boldsymbol{\lambda}_k$ over the impact time. The impact is completely elastic for $\epsilon_k = 1$ and inelastic for $\epsilon_k = 0$. In this way, each normal cone inclusion can be equipped with an impact inclusion similar to Eqn. (5). Strictly speaking, the set $\mathcal{D}_k$ is the integrated convex set $\mathcal{C}_k$ over the impact time. However, for unilateral contacts with spatial Coulomb friction and Moreau's discretization scheme, the convex sets remain the same, in the sense that $\mathcal{D}_N = \mathcal{C}_N = \mathbb{R}_0^-$, $\mathcal{D}_T = \mathcal{C}_T(\mu\Lambda_N)$, and $\mathcal{D}_B = \mathcal{C}_B = \mathbb{R}^d$. For bilateral constraints, a completely inelastic impact law with a zero restitution coefficient is assumed.

Moreau's Discretization Scheme

The discretization of Moreau was firstly introduced in [14] and is a difference scheme to evaluate the equations of motion with set-valued contact laws in Eqn. (1) together with both the impact and impact-free motion. The Moreau discretization scheme is an explicit time-stepper composed of a midpoint discretization on displacement level and an Euler backward method on velocity level. Moreau's time stepping scheme over a time step $\Delta t = t^E - t^S$ from $(\mathbf{q}^S, \mathbf{u}^S)$ to new state $(\mathbf{q}^E, \mathbf{u}^E)$ is given as

First semi time step: $\quad \mathbf{q}^M = \mathbf{q}^E + \frac{1}{2}\Delta t \mathbf{F}(\mathbf{q}^S)\mathbf{u}^S \quad (6)$

Solve inclusion problem for $\mathbf{u}^E, \boldsymbol{\Lambda}_U, \boldsymbol{\Lambda}_B$:

$$\mathbf{M}\,(\mathbf{u}^E - \mathbf{u}^S) - \mathbf{h}\Delta t - \mathbf{W}_U\boldsymbol{\Lambda}_U - \mathbf{W}_B\boldsymbol{\Lambda}_B = \mathbf{0}$$

$$\left.\begin{array}{c} \boldsymbol{\gamma}_{U_i}^S = \mathbf{W}_{U_i}^\top \mathbf{u}^S, \quad \boldsymbol{\gamma}_{U_i}^E = \mathbf{W}_{U_i}^\top \mathbf{u}^E \\ \boldsymbol{\xi}_{U_i} := \boldsymbol{\gamma}_{U_i}^E + \boldsymbol{\epsilon}_i \boldsymbol{\gamma}_{U_i}^S \\ (\boldsymbol{\xi}_{U_i}, \boldsymbol{\Lambda}_{U_i}) \in \mathcal{U}_i \end{array}\right\} \forall i \in \mathcal{I}(\mathbf{q}), \quad (7)$$

$$\left.\boldsymbol{\gamma}_{B_j}^E = \mathbf{W}_{B_j}^\top \mathbf{u}^E, \quad \boldsymbol{\xi}_{B_j} = \boldsymbol{\gamma}_{B_j}^E, \quad (\boldsymbol{\xi}_{B_j}, \boldsymbol{\Lambda}_{B_j}) \in \mathcal{B}_j \right\} \forall j.$$

Second semi time step: $\quad \mathbf{q}^E = \mathbf{q}^M + \frac{1}{2}\Delta t \mathbf{F}(\mathbf{q}^M)\mathbf{u}^E. \quad (8)$

The mass matrix $\mathbf{M}(\mathbf{q})$, the term $\mathbf{h}(\mathbf{q}, \mathbf{u})$ and all generalized force directions $\mathbf{W}(\mathbf{q})$ are evaluated at the midpoint $(\mathbf{q}^M, \mathbf{u}^S)$. For numerical reasons the set $\mathcal{I}$ is now changed to the set of all contact indices i with $g_{N_i}(\mathbf{q}) \leq 0$. The i-th set-valued contact force laws in Eqn. (2) and Eqn. (3) for the *hard contact* model and the set-valued force law for the bilateral constraints in Eqn. (4) are gathered in the set $\mathcal{U}_i$ and $\mathcal{B}_j$ in Eqn. (7) respectively as

$$\mathcal{U}_i := \left\{ (\boldsymbol{\xi}_i, \boldsymbol{\Lambda}_i) \mid \xi_{N_i} \in \mathcal{N}_{\mathbb{R}_0^-}(-\Lambda_{N_i}), \boldsymbol{\xi}_{T_i} \in \mathcal{N}_{\mathcal{D}_{T_i}}(-\boldsymbol{\Lambda}_{T_i}) \right\} \quad (9)$$

$$\mathcal{B}_j := \left\{ (\boldsymbol{\xi}_j, \boldsymbol{\Lambda}_j) \mid \boldsymbol{\xi}_j \in \mathcal{N}_{\mathbb{R}^{d_j}}(-\boldsymbol{\Lambda}_j) \right\}. \quad (10)$$

The diagonal matrix of the restitution coefficients $\boldsymbol{\epsilon}_i$ in Eqn. (7) is defined as $\boldsymbol{\epsilon}_i := \mathrm{diag}(\epsilon_{N_i}, \epsilon_{T_i}, \epsilon_{T_i}) \in \mathbb{R}^{3\times 3}$. The compliant contact model does not include impulsive forces and thus no impact law can be formulated ($\boldsymbol{\Lambda}_U$ vanishes in Eqn. (7)). The contribution of the impressed compliant forces $\boldsymbol{\lambda}_U$ to the term $\mathbf{h}$ can be computed as described in Alg. 1 with the generalized coordinates $\mathbf{q}^M$ and velocities $\mathbf{u}^S$. The inclusion problem in Eqn. (7) is the starting point for the derivation of the proximal point equations in the next section which then can be solved with the iteration schemes explained later in this paper.

Proximal Point Equation

In [20], it has been shown that each normal cone inclusion to a convex set $\mathcal{C}$ can be transformed into a proximal point equation to the set $\mathcal{C}$ as

$$\mathbf{y} \in \mathcal{N}_{\mathcal{C}}(\mathbf{x}) \quad \Leftrightarrow \quad \mathbf{x} = \mathrm{prox}_{\mathcal{C}}^{R}(\mathbf{x} + \mathbf{R}^{-1}\mathbf{y}), \tag{11}$$

The superscript R denotes the norm defined as $||\mathbf{x}||_R = \sqrt{\mathbf{x}^\top \mathbf{R}\mathbf{x}}$, where $\mathbf{R}$ is a symmetric positive definite matrix. The proximal function $\mathbf{y} = \mathrm{prox}_{\mathcal{C}}^{R}(\mathbf{x})$ projects its argument $\mathbf{x}$ to the set $\mathcal{C}$ in a way that the norm $||\mathbf{y} - \mathbf{x}||_R$ attains the minimum.

By using the scaled Euclidean norm $\mathbf{R} = 1/r\mathbf{I}, r \in \mathbb{R}^+$ to simplify the projections, all normal cone inclusions in Eqn. (7) can be rewritten as proximal point equations which then yields the following set of equations

$$\boldsymbol{\xi} = \mathbf{G}\boldsymbol{\Lambda} + \mathbf{c}, \tag{12}$$

$$\left\{ (\boldsymbol{\xi}_{U_i}, \boldsymbol{\Lambda}_{U_i}) \,\middle|\, \begin{array}{l} \Lambda_{N_i} = \text{-}\mathrm{prox}_{\mathbb{R}_0^-}(\text{-}\Lambda_{N_i} + r_{N_i}\xi_{N_i}) \\ \boldsymbol{\Lambda}_{T_i} = \text{-}\mathrm{prox}_{\mathcal{D}_{T_i}}(\text{-}\boldsymbol{\Lambda}_{T_i} + r_{T_i}\boldsymbol{\xi}_{T_i}) \end{array} \right\} \forall i \in \mathcal{I} \tag{13}$$

$$\boldsymbol{\Lambda}_{B_j} = \text{-}\mathrm{prox}_{\mathbb{R}^{d_j}}(\text{-}\boldsymbol{\Lambda}_{B_j} + \mathrm{diag}(r_{B_{j,1}}, \ldots, r_{B_{j,d_j}})\boldsymbol{\xi}_{B_j}) \;\forall j \tag{14}$$

$$\begin{aligned} \mathbf{G} &:= \mathbf{W}^\top \mathbf{M}^{-1}\mathbf{W}, \quad \mathbf{W} := [\mathbf{W}_U, \mathbf{W}_B] \\ \mathbf{c} &:= \mathbf{W}^\top \mathbf{M}^{-1}\mathbf{h}\Delta t + (\mathbf{I} + \boldsymbol{\epsilon})\mathbf{W}^\top \mathbf{u}^S \\ \boldsymbol{\Lambda} &:= [\boldsymbol{\Lambda}_U^\top, \boldsymbol{\Lambda}_B^\top]^\top, \quad \boldsymbol{\xi} := [\boldsymbol{\xi}_U^\top, \boldsymbol{\xi}_B^\top]^\top \\ \boldsymbol{\epsilon} &:= \mathrm{diag}(\boldsymbol{\epsilon}_U, \boldsymbol{\epsilon}_B), \\ \boldsymbol{\epsilon}_U &:= \mathrm{diag}(\ldots, \boldsymbol{\epsilon}_i, \ldots), \quad \boldsymbol{\epsilon}_B := \mathrm{diag}(\mathbf{0}, \ldots, \mathbf{0}) \end{aligned} \tag{15}$$

These equations can be obtained by gathering all vectors associated with a contact i and bilateral constraint j and substituting all definitions in Eqn. (7) into $\boldsymbol{\xi}_{U_i}$ and $\boldsymbol{\xi}_{B_j}$. The importance of the scalar values r in each proximal equation is discussed in the next section where we introduce two commonly used iteration schemes to solve Eqn. (12-14) for $\boldsymbol{\Lambda}$ in each time step of the Moreau scheme in Eqn. (6-8). One should note that the projection $\mathrm{prox}_{\mathbb{R}^{d_j}}$ in Eqn. (14) is the identity map and can therefore be simplified.

JOR and SOR Prox Iteration

The projected over-relaxed Jacobi (JOR Prox) and Gauss-Seidel (SOR Prox) iteration in [12, 20] are the two main iteration schemes that are investigated in this paper. Each proximal equation in Eqn. (12, 13, 14) can be converted into a fixed-point iteration with iteration index k in the form

$$\mathbf{x}^{k+1} = -\mathrm{prox}_{\mathcal{C}}(-\mathbf{x}^k + r\boldsymbol{\xi}(\mathbf{x}^k)), \quad r > 0 \tag{16}$$

By inserting Eqn. (12) into (13, 14) a projective iteration scheme in the impulsive forces $\boldsymbol{\Lambda}$ is obtained. Given an initial $\boldsymbol{\Lambda}^0 = [\boldsymbol{\Lambda}_U^{0\top}, \boldsymbol{\Lambda}_B^{0\top}]^\top$, the JOR Prox scheme iterates over all contacts i and bilateral contacts j and computes the new values according to Eqn. (16). The SOR Prox scheme does the same but reuses the already projected new values in vector $\boldsymbol{\Lambda}^{k+1}$ for the projection of the successive value(s). The parameters $r_{N_i}, r_{T_i}\ \forall i$ and $(r_{B_{j,1}}, \ldots, r_{B_{j,d_j}})\ \forall j$ can be chosen as $\alpha/\mathbf{G}_{ll}$ where $\mathbf{G}_{ll}$ is the diagonal value on the corresponding row of the symmetric Delassus matrix $\mathbf{G}$. The tangential direction uses only one parameter r_{T_i}, therefore the maximum of the two diagonal values in $\mathbf{G}$ is chosen. With this choice, the iteration scheme mentioned is similar to a Jacobi or Gauss-Seidel iteration for the linear system $\mathbf{G}\boldsymbol{\Lambda} + \mathbf{c} = \mathbf{0}$. The relaxation parameter $\alpha \in (0, 2)$ mentioned above determines the convergence behaviour of the JOR and SOR Prox scheme. The iteration schemes are aborted if all components $\boldsymbol{\Lambda}_{(i)}$ of $\boldsymbol{\Lambda}$ are converged and fulfill the following tolerance criteria in the form $|\boldsymbol{\Lambda}_{(i)}^{k+1} - \boldsymbol{\Lambda}_{(i)}^{k}| \le |\boldsymbol{\Lambda}_{(i)}^{k}|\, T_{rel} + T_{abs} \quad \forall i$. The scalars T_{rel} and T_{abs} are the relative and absolute tolerance. This termination criterion controls the absolute error if $|\boldsymbol{\Lambda}_{(i)}^{k}| \ll 1$, and the relative error otherwise.

Calculating the mostly sparse symmetric Delassus matrix $\mathbf{G}$ in Eqn. (12) is cumbersome and has a large memory footprint. The calculation can be avoided with an equivalent fixed-point scheme which iterates in the end-time velocities $\mathbf{u}^{E,k}$ (superscript E neglected in the following). It can be obtained by inserting the definition for $\boldsymbol{\xi}_{U_i}$ in Eqn. (12) into Eqn. (13) as

$$\mathbf{u}^k := \mathbf{u}^S + \mathbf{M}^{-1}(\mathbf{h}\Delta t + \mathbf{W}_U\boldsymbol{\Lambda}_U^k + \mathbf{W}_B\boldsymbol{\Lambda}_B^k) \tag{17}$$

with

$$\left. \begin{aligned} \boldsymbol{\Lambda}_{U_i}^{k+1} &= -\mathrm{prox}_{[U_i]}\big(\text{-}\boldsymbol{\Lambda}_{U_i}^k + \mathbf{R}_{U_i}^{-1}(\mathbf{W}_{U_i}\mathbf{u}^k + \mathbf{b}_i)\big) \\ \mathbf{u}^{k+1} &= \mathbf{u}^k + \mathbf{M}^{-1}\mathbf{W}_{U_i}(\boldsymbol{\Lambda}_{U_i}^{k+1} - \boldsymbol{\Lambda}_{U_i}^k) \end{aligned} \right\} \forall i \tag{18}$$

respectively

$$\left. \begin{aligned} \boldsymbol{\Lambda}_{B_j}^{k+1} &= \boldsymbol{\Lambda}_{B_j}^k - \mathbf{R}_{B_j}^{-1}\mathbf{W}_{B_j}\mathbf{u}^k \\ \mathbf{u}^{k+1} &= \mathbf{u}^k + \mathbf{M}^{-1}\mathbf{W}_{B_j}(\boldsymbol{\Lambda}_{B_j}^{k+1} - \boldsymbol{\Lambda}_{B_j}^k) \end{aligned} \right\} \forall j \tag{19}$$

where vector $\mathbf{b}_i := \boldsymbol{\epsilon}_i\mathbf{W}_{U_i}\mathbf{u}^S$ and $\mathrm{prox}_{[U_i]}$ vectorially collects all proximal functions for contact i. The diagonal matrices $\mathbf{R}_{U_i}$ and $\mathbf{R}_{B_j}$ contain all r values for each prox function. The JOR Prox scheme is obtained if first all impulsive forces are projected in (18-19) and afterwards all velocity updates $\mathbf{u}^{k+1}$ are computed. In contrast, the SOR Prox scheme is obtained by applying first Eqn. (18) and afterwards Eqn. (19). This corresponds to a succession pattern that evaluates each contact successively. The projective succession pattern can be even more fine-grained if the velocity updates are computed after each individual projection at each contact i (cf. (13), normal and tangential direction).

Instead of iterating over the bilateral constraints in Eqn. (19), the bilateral percussions can be calculated directly for a given $\boldsymbol{\Lambda}_U^k$ by evaluating $\boldsymbol{\gamma}_B^E = \mathbf{0}$ at the price of a matrix inversion

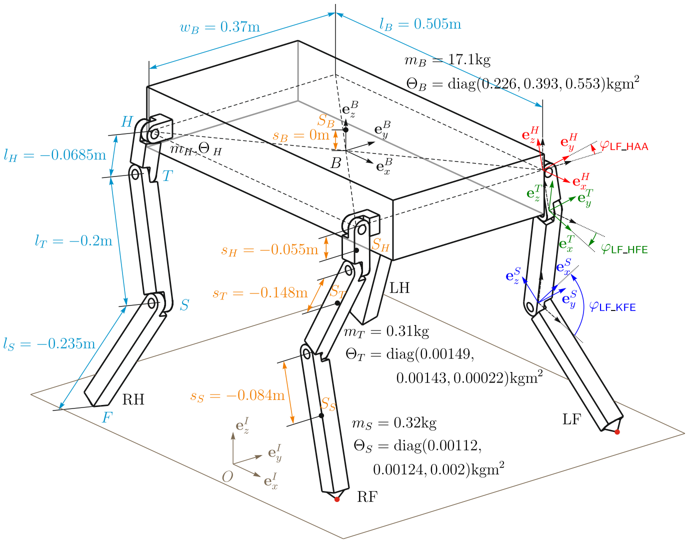


FIGURE 4: MECHANICAL MODEL OF THE QUADRUPED

which yields

$$\boldsymbol{\Lambda}_B = -\mathbf{G}_B^{-1}\mathbf{W}_B^\top\left(\mathbf{u}^S+\mathbf{M}_M^{-1}(\mathbf{h}_M\Delta t+\mathbf{W}_U\boldsymbol{\Lambda}_U^k)\right), \tag{20}$$

where the bilateral Delassus matrix $\mathbf{G}_B := \mathbf{W}_B^\top\mathbf{M}^{-1}\mathbf{W}_B$ is used. By using the termination criterion mentioned above, the JOR or SOR Prox can still cycle between different feasible solutions. This has not been experienced with our mechanical system so far. In the case of cycling, a termination criterion in the kinetic energy $\frac{1}{2}\mathbf{u}^\top\mathbf{M}\mathbf{u}$ would be a more appropriate measure as the velocities tend to be unique (apart from problems like the Painlevé paradox). A discussion of the convergence of the JOR and SOR Prox schemes can be found in [12, 20].

DESCRIPTION OF THE QUADRUPEDAL ROBOT

The model of the symmetric quadruped StarlETH is illustrated in Fig. 4. The 13 rigid bodies are connected by 12 actuated hinge joints with angles φ_i. The legs are attached to the trunk at the hip abduction/adduction (HAA) joints. A unilateral contact with Coulomb friction is introduced at each foot and joint to prevent the robot from sinking into an even and flat ground.

A right-handed coordinate system that is fixed to the ground, the inertial frame I, and a body-fixed system for each of the 13 bodies are introduced as indicated in Fig. 4 to describe the system. We choose a Hamiltonian unit quaternion $P = p_0 + p_1 i + p_2 j + p_3 k \in \mathbb{H}$ as parameterization of a rotation to represent the orientation of a body [21]. The coefficients of the quaternion are stored in the vector $\mathbf{p}_{BI} = \left[p_0, p_1, p_2, p_3\right]^\top \in \mathbb{R}^4$ to determine a rotation from frame I to B. The benefit of quaternions is that they do not have any singularities like Euler angles, which is important for legged robots that can turn over. However, the unit-length constraint needs to be considered in the equations of motion.

Since the choice of the generalized coordinates $\mathbf{q}$ and velocities $\mathbf{u}$ affects the performance of the simulation in terms of accuracy and speed, we investigate three different sets of coordinates: the so-called minimal, extended and maximal set of coordinates.

The minimal set of coordinates and velocities of the quadruped are

$$\begin{aligned}\mathbf{q} &= \left[{}_I\mathbf{r}_{OB}^\top,\ \mathbf{p}_{BI}^\top,\ \varphi_1, \cdots, \varphi_{12}\right]^\top \in \mathbb{R}^{19},\\ \mathbf{u} &= \left[{}_I\mathbf{v}_B^\top,\ {}_B\boldsymbol{\omega}_{IB}^\top,\ \dot{\varphi}_1, \cdots, \dot{\varphi}_{12}\right]^\top \in \mathbb{R}^{18},\end{aligned} \tag{21}$$

where ${}_I\mathbf{r}_{OB} \in \mathbb{R}^3$ is the position of the trunk with respect to the origin of the inertial frame expressed in the inertial frame, $\mathbf{p}_{BI} \in \mathbb{R}^4$ is the unit quaternion that describes the orientation of the trunk with respect to the inertial frame, and $\varphi_i \in \mathbb{R}$ are the joint angles. The generalized velocities $\mathbf{u}$ are the linear velocity of the trunk ${}_I\mathbf{v}_B$ expressed in the inertial frame, the angular velocity of the trunk ${}_B\boldsymbol{\omega}_{IB}$ expressed in the body-fixed frame B, and the time derivatives of the joint angles $\dot{\varphi}_i$.

The time derivative of the generalized coordinates $\dot{\mathbf{q}}$ is not equal to the generalized velocities $\mathbf{u}$, because we use the angular velocity ${}_B\boldsymbol{\omega}_{IB}$ instead of the time derivative of the quaternion $\dot{\mathbf{p}}_{BI} = \frac{1}{2}\bar{\mathbf{H}}^\top(\mathbf{p}_{BI})_B\boldsymbol{\omega}_{IB}$. The matrix $\mathbf{F} \in \mathbb{R}^{19\times18}$ in Eqn. (1) for the minimal set of coordinates has therefore the form

$$\begin{aligned}\mathbf{F} &= \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \frac{1}{2}\bar{\mathbf{H}}^\top(\mathbf{p}_{BI}) & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix}, \quad \hat{\mathbf{p}}_{1:3} = \begin{bmatrix}0 & -p_3 & p_2\\ p_3 & 0 & -p_1\\ -p_2 & p_1 & 0\end{bmatrix},\\ \bar{\mathbf{H}} &= \left[-\mathbf{p}_{1:3},\ -\hat{\mathbf{p}}_{1:3} + p_0\mathbf{I}_{3\times3}\right] \in \mathbb{R}^{3\times4}.\end{aligned} \tag{22}$$

The maximal set of coordinates directly describes the absolute position of the center of mass and the orientation of each body instead of describing them relative to each other with the joint angles. This results in a system with 78 DOF and therefore needs 60 bilateral constraints, but benefits from constant mass matrices which need to be inverted only once.

By cutting the lightweight legs free from the heavy trunk at the HAA joints, the extended set of coordinates may be superior than the other two extreme descriptions. The resulting five subsystems have in total 38 DOF, which need to be reduced to 18 by adding 20 bilateral constraints at the HAA joints to the equations of motion.

EVALUATION

A locomotion controller needs to be active during the experiments for the evaluation, primarily because legged robots are

TABLE 1: EVALUATED SIMULATION CONFIGURATIONS

Name	Contact model	Generalized Coordinates	Prox scheme	Bilateral constraints	α	Evaluations per second	Realtime factor
Cm	Compliant	minimal	-	-	-	8’900	11.5
HmJ	**Hard**	**minimal**	**JOR**	**-**	**0.6**	**5’400**	**13.5**
HmS	Hard	minimal	SOR	-	0.6	4’700	11.8
HEJD	Hard	Extended	JOR	Direct	0.6	2’700	6.8
HEJP	Hard	Extended	JOR	Prox	0.5	2’900	7.1
HESD	Hard	Extended	SOR	Direct	0.6	2’500	6.1
HESP	Hard	Extended	SOR	Prox	0.6	2’400	6.0
HMJD	Hard	Maximal	JOR	Direct	1.0	1’600	4.1
HMJP	Hard	Maximal	JOR	Prox	0.4	600	1.5
HMSD	Hard	Maximal	SOR	Direct	1.0	1’600	3.9
HMSP	Hard	Maximal	SOR	Prox	0.6	800	2.0

inherently unstable. We employ a model-based locomotion controller [22] which demonstrated robust walking and trotting in the presence of external perturbations. The quadruped is either maintaining a default posture similar to the one shown in Fig. 1 or trotting on flat and even ground, where the two diagonal pairs of legs are alternately in contact with the ground with a gait cycle of 1.25 Hz.

The controller runs with a constant control update rate of 400 Hz and defines therefore an upper limit for the maximal length of the simulation time step ($N \cdot \Delta t = 2.5$ ms, with $N \in \mathbb{N}^+$).

The usage of a controller comes along with two problems: Firstly, the controller has a variety of control parameters that need to be tuned and therefore a coupling between the performance of the controller and the simulation exists. Secondly, the controller is able to deal with perturbations, which may blur the performance of the simulation. Nevertheless, we tune the control parameters once with the compliant contact models and keep them constant during all experiments.

Table 1 gives an overview of the different simulation configurations that are evaluated. The type of contact models, the choice of the generalized coordinates, the prox schemes and the calculation methods of the bilateral impulses are compared with each other.

The focus of the evaluation clearly lies on the computation effort, which is very important for robotics. Therefore, all algorithms are implemented in C++11 for comparison. The computations of the kinematic and dynamic quantities such as Jacobians or the mass matrix utilize pre-generated code. The equations are symbolically optimized with the Symbolic Toolbox of Matlab with our open-source toolbox proNEu [23]. The tool derives the dynamics from the projected Newton-Euler equations from a kinematic tree and was extended with quaternions for this work. The matrix inversions are computed with a standard Cholesky decomposition provided by the Eigen C++ library [24]. To evaluate the speeds, the computation time is measured using the posix time from the Boost C++ libraries [25], and an Intel Xeon E31245 processor (3.3 GHz).

The friction parameter ($\mu = 0.8$) and restitution parameters ($\epsilon_T = \epsilon_N = 0.0$) of the contact models are kept constant during all experiments.

Robustness and Ease of Use

The robustness of a simulation needs to be guaranteed against floating-point rounding errors, convergence problems, chattering effects and so forth. It is a measure for the physical and numerical reliability of the simulation in all different kinds of situations. As a stress test, we pushed the robot while standing at the main body such that it fell on its side. This generated high impacts and multiple closed contacts.

In most cases, accuracy and speed can be influenced by looser tolerances and faster computers, the robustness however depends mainly on the chosen simulation model and its parameters.

Compliant contact models require manual tuning of a couple of parameters; five for the introduced model. This tedious process depends on the appearing contact forces which are a function of the kinematic and dynamic properties as well as the behavior of the robot, and thus may be initially unknown. The stiffness of the springs should be as high as possible for high accuracy, but there is an upper limit where the differential equations become too stiff. Adjusting the simulation step size may help to address this numerical problem to some extent, but adversely affects the computation effort. We found the following parameter values for a robust simulation with two updates per control update: $c_N = c_D =$ 30’000 N/m, $d_N = d_T =$ 50 N s/m.

In case of the proposed hard contact model, the physical parameters are well defined by the mechanical system and only the optimal value of the numerical parameter α together with the termination criteria of the iterations need to be found. α is a measure of the “aggressivity” of the iterative calculation. The simulation speed increases with this value, but if it exceeds a critical limit, the iteration may not converge anymore. A conservative value should be chosen if speed is less critical. As the differential equations are of first order and not very stiff, larger step sizes than for compliant contact models are possible without having any great influence on the robustness.

In conclusion, the hard contact model has a clear advantage over the compliant model in terms of ease of use and robustness, because it is easier to find good parameter values, although the implementation of the simulation is more challenging.

Accuracy

Both contact models have their drawbacks in terms of accuracy. The feet of the robot while trotting sink about 4.4 mm into the ground if the compliant contacts are active, because the springs cannot be chosen stiffer for numerical reasons. This value is very high compared to the deformability of foot and ground. The hard contact simulation on the other side has to deal with the drift phenomenon, which occurs at unilateral and bilateral contacts. The tolerances $T_{rel} = T_{abs} = 10^{-6}$ used in the iteration schemes yield acceptable drifts: In standing position, the vertical drift of the unilateral foot constraints is about 750 µm/min independently from the step size. However the unilateral constraint violations are reset periodically in any case during locomotion, such that this is not a severe issue. The drift of the bilateral constraints while trotting is about 0.59 µm/min, which is negligible.

Due to the different nature of the two contact models and missing ground truth, a quantitative comparison is very difficult. A possible method is to investigate the contact forces. Since the hard contact laws output impulses instead of forces, we compute the force $\mathbf{F} = \mathbf{\Lambda}_U/\Delta t$ averaged over one simulation time step for comparison. Figure 5 shows the normal component of the computed contact force of the left-fore foot during one cycle of a trotting gait. The resulting contact force of the compliant and the hard contact model coincide well. The low damping of the evaluated compliant contact model (Cm) leads to oscillations (solid blue curve). If the damping is increased by a factor of ten, the oscillations are damped out rapidly, however impacts generate higher forces and the step size has to be decreased by a factor of four.

Our experience shows that the hard contact model is preferable concerning the physical correctness of the simulation.

Computation Speed

The computational efforts of the various simulation configurations depend on one side on the calculation of the smooth dynamics, i.e. the computation of the mass matrix $\mathbf{M}$ and the force vector $\mathbf{h}$, and on the computational complexity of the contact models on the other side. While the computation cost of the compliant contact laws is low and the calculation is straightforward, the effort needed for the hard contacts may be reduced substantially with the right choice of the numerical iteration algorithm. We therefore analyze the influence on the number of iterations needed to solve the constitutive laws of the hard contact model first. As a second step we investigate the computation effort needed for one simulation step and finalize the evaluation by examining the required number of steps and size of time step needed to optimize the simulation speed while guaranteeing robustness and accuracy.

Number of Hard Contact Iterations The number of required prox iterations for one time step mainly depends on the relaxation parameter α and the convergence abort criterion. Since the relative and absolute tolerances of the convergence abort criterion are determined by the desired accuracy as aforementioned, we analyze the correlation between α and the number of iterations. The results are depicted in Fig. 6.

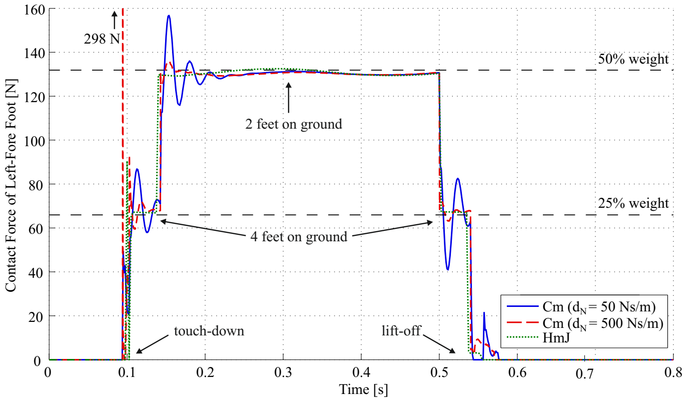


FIGURE 5: NORMAL COMPONENT OF CONTACT FORCE

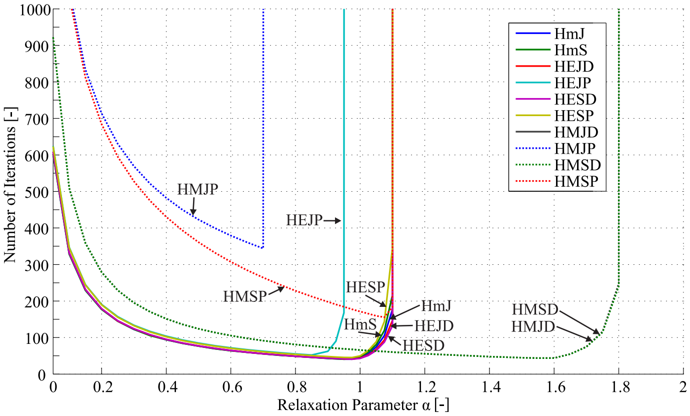


FIGURE 6: NUMBER OF PROX ITERATIONS

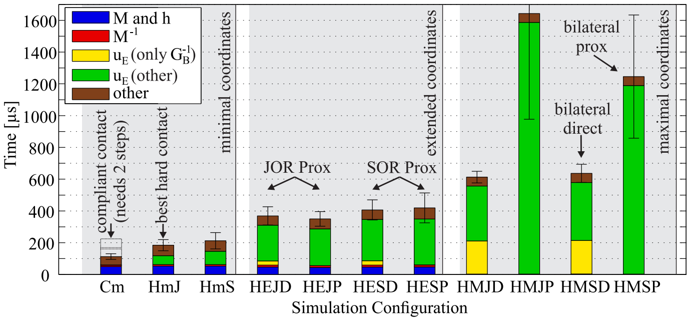


FIGURE 7: COMPUTATION TIME FOR ONE TIME STEP

The general advantage of the SOR Prox over the JOR is faster and more reliable convergence. This nature is demonstrated by the results of HMJP and HMSP in Fig. 6. Surprisingly, this effect does not appear for the configuration with minimal coordinates. In this case, the SOR Prox updates the velocity of the entire robot after updating the contact impulse of a foot contact.

This method only has a benefit if the impulse of one contact affects the velocities of the other contacts, otherwise it may cause useless, but costly velocity updates within one iteration. Numerically this can be explained by investigating the unilateral Delassus matrix $\mathbf{G}_U = \mathbf{W}_U^\mathsf{T}\mathbf{M}^{-1}\mathbf{W}_U$, which is the factor between the contact impulses acting on the closed unilateral contacts and the change in velocity of all those contact points. For the quadruped described by the minimal set of coordinates, the matrix $\mathbf{G}_U$ is block diagonal dominant, when the knees are bent. As a consequence, a contact impulse on one foot will mostly influence its own velocity and has mainly an effect on the lightweight legs than on the heavy trunk, and so the momentum is not propagated through the entire system.

For every configuration, there is an upper limit of α where the prox iteration stops to converge. The fastest convergence is obtained with an α close to this limit as shown in Fig. 6. Since the upper limit varies for different gaits, a high α does not guarantee robustness in all situations. As a good compromise between robustness and speed, we choose α to be half of the upper limit for further evaluations. The selected values are listed in Table 1 and allow the robot to be disturbed by large external forces.

Computation Effort for One Time Step The computation time of one simulation step is broken down to the calculation time of $\mathbf{M}$ and $\mathbf{h}$, the inversion of $\mathbf{M}$, the calculation and inversion of $\mathbf{G}_B$ and the calculation of $\mathbf{u}_E$. Since the calculation effort varies over the gait cycle, in particular for the hard contact simulation, the means and standard deviations over 20 s (25 gait cycles) were measured.

The results are visualized in Fig. 7 and listed in Table 1. The calculation costs of $\mathbf{M}$, $\mathbf{h}$ and the inversion of $\mathbf{M}$ depend only on the choice of coordinates. The description with a minimal set of coordinates with a large mass matrix needs more computation time for $\mathbf{M}$, $\mathbf{h}$ and $\mathbf{M}^{-1}$ as the one with a maximal description as expected. Compared to the configuration with the minimal set of coordinates, the one with the extended set of coordinates does not show any gain in speed.

The time fraction for the computation of $\mathbf{u}_E$ is very small as expected for the compliant contacts. The computation time of $\mathbf{u}_E$ mostly depends on the set of coordinates as well as on the iteration method for the hard contacts. The higher the amount of coordinates, the more constraints have to be introduced which requires more computation effort.

For small subsystems with many bilateral constraints, the SOR Prox starts to outperform the JOR Prox. However if the minimal set of coordinates is chosen, the JOR Prox leads to better results. The schemes need a similar amount of iterations, but one JOR iteration is cheaper.

The computation time required to obtain the bilateral impulses either includes an iteration or the calculation of $\mathbf{G}_B^{-1}$. For the configuration with the extended set of coordinates both methods perform similarly. If the number of bilateral constraints increases, the direct calculation becomes beneficial, although $\mathbf{G}_B$ becomes very large.

The remaining calculation time is spent on the collision detection and data management, which is not affected by the simulation configuration in our case.

The standard deviation is larger for the hard contact simulations, because the effort for the prox iterations increases with the number of closed contacts which varies over one gait period. The calculation time for the compliant contact simulation however is not much affected by this number.

Overall Computation Effort Minimal computation effort is achieved when the simulation updates only once per control update and the simulation time step Δt is equal to the control time step. The simulation with compliant contacts needs two steps per control update at minimum for numerical stability as opposed to the simulation with hard contacts that is stable with one step. For this reason, the simulation with hard contacts overall performs faster, although one step needs more computation time then one for the compliant contacts.

The comparison of the computation time between the simulation with compliant and hard contacts have to be treated with caution. We have only investigated a very simple compliant model. More complex models with nonlinear stiffness and damping may allow larger time steps. The computation time of $\mathbf{u}_E$ may be increased drastically for compliant contacts by applying the Articulated-Rigid-Body algorithm [1], which does not require the calculation of the mass matrix and its inversion explicitly.

In summary, the simulations with hard and compliant contact model lead to comparable performance.

CONCLUSION

A simulation technique based on Moreau's time-stepping scheme was presented and evaluated for a quadrupedal robot. We compared the proposed method with a simulation method using a compliant contact model which was previously used for the development of locomotion control algorithms.

Compliant contact models are widely used in robotics because of their low computation time. However, the experimental results indicate that the proposed method with hard contacts can be superior in terms of speed despite its complexity.

The description with minimal coordinates in combination with the JOR Prox scheme led to the best results regarding computation speed. Surprisingly, the advanced SOR algorithm performs slower due to the mechanical structure of the multibody system.

The hard contact simulation returns more reasonable results in demanding situations as long as the relaxation parameter α is adequately chosen. If it is too "aggressive", the numerical iteration does not converge in every situation, which is a drawback of the hard contact model. The adjustment of the hard contact parameters is very simple as opposed to the compliant contact parameters, which are more of numerical than physical nature.

In conclusion, the proposed simulation method is very beneficial for a quadrupedal robot. To tranfer these results to other multi-legged robots with many more degrees-of-freedom, the scalability of the method should be addressed in future work.

ACKNOWLEDGMENT

This research was supported by the Swiss National Science Foundation through the National Centre of Competence in Research Robotics.

REFERENCES

[1] Featherstone, R., 2008. *Rigid body dynamics algorithms*. Springer, New York.

[2] Marhefka, D., and Orin, D., 1999. "A compliant contact model with nonlinear damping for simulation of robotic systems". *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* ***29***(6), pp. 566–572.

[3] Bibalan, P. T., and Featherstone, R., 2009. "A Study of Soft Contact Models in Simulink". In Australasian Conference on Robotics and Automation, no. 1.

[4] Drumwright, E., Hsu, J., Koenig, N., and Shell, D., 2010. "Extending open dynamics engine for robotics simulation". In *Simulation, Modeling, and Programming for Autonomous Robots*, O. Ando, Noriaki and Balakirsky, Stephen and Hemker, Thomas and Reggiani, Monica and Stryk, ed., Vol. 6472 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, pp. 38–50.

[5] Coumans, E., Bullet Physics Library, January, 2013. http://bulletphysics.org.

[6] Michel, O., 2004. "Webots: Professional Mobile Robot Simulation". *Int. Journal of Advanced Robotic Systems,* ***1***(1), pp. 39–42.

[7] Calderon, C., Mohan, R., and Zhou, C., 2008. "Virtual-re: A humanoid robotic soccer simulator". In Int. Conf. on Cyberworlds, pp. 561–566.

[8] Todorov, E., Erez, T., and Tassa, Y., 2012. "MuJoCo: A physics engine for model-based control". *IEEE/RSJ Int. Conf. on Int. Robots and Systems*, Oct., pp. 5026–5033.

[9] Kanehiro, F., Fujiwara, K., Kajita, S., Yokoi, K., Kaneko, K., Hirukawa, H., Nakamura, Y., and Yamane, K., 2002. "OpenHRP: Open Architecture Humanoid Robotics Platform". In IEEE Int. Conf. on Robotics and Automation, Vol. 1, pp. 24–30.

[10] Dallali, H., Mosadeghzad, M., Medrano-Cerda, G., Docquier, N., Kormushev, P., Tsagarakis, N., Li, Z., and Caldwell, D., 2013. "Development of a dynamic simulator for a compliant humanoid robot based on a symbolic multibody approach". In Mechatronics (ICM), 2013 IEEE Int. Conf. on, pp. 598–603.

[11] Stewart, D. E., and Trinkle, J. C., 1996. "An Implicit Time-Stepping Scheme For Rigid Body Dynamics With Inelastic Collisions and Coulomb Friction". *Int. Journal for Numerical Methods in Engineering,* ***39***(15), pp. 2673–2691.

[12] Möller, M., 2011. "Consistent Integrators for Non-Smooth Dynamical System". PhD thesis, ETH Zürich. No. 19715.

[13] Transeth, A., Leine, R., Glocker, C., Pettersen, K., and Liljeback, P., 2008. "Snake robot obstacle-aided locomotion: Modeling, simulations, and experiments". *Robotics, IEEE Transactions on,* ***24***(1), pp. 88–104.

[14] Moreau, J., 1988. "Unilateral contact and dry friction in finite freedom dynamics". In *Nonsmooth Mechanics and Applications*, J. Moreau and P. Panagiotopoulos, eds., Vol. 302 of CISM, Courses and Lectures. Springer, Vienna.

[15] Hutter, M., Gehring, C., Bloesch, M., Hoepflinger, M., Remy, C. D., and Siegwart, R., 2012. "StarlETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion". In Proc. of the Int. Conf. on Climbing and Walking Robots, pp. 1–8.

[16] Schaal, S., 2009. The SL Simulation and Real-Time Control Software Package. Tech. rep., University of Southern California, USA.

[17] Glocker, Ch., 2013. "Simulation of Hard Contacts with Friction: An Iterative Projection Method". In *Recent Trends in Dynamical Systems*, A. Johann, H.-P. Kruse, F. Rupp, and S. Schmitz, eds. Springer Proceedings in Mathematics & Statistics, Vol. 35, pp. 493–513.

[18] Glocker, Ch., 2001. *Set-Valued Force Laws, Dynamics of Non-Smooth Systems*, Vol. 1 of *Lecture Notes in Applied Mechanics*. Springer-Verlag, Berlin.

[19] Glocker, Ch., 2006. "An introduction to impacts". In *Nonsmooth mechanics of solids*. Springer, pp. 45–101.

[20] Studer, C., 2009. *Numerics of unilateral contacts and friction*. Lecture Notes in Applied and Computational Mechanics 47. Berlin: Springer. ix.

[21] Hamilton, W. R., 1844. "On Quaternions, or on a New System of Imaginaries in Algebra". *Philosophical Magazine,* ***25***(3), pp. 489–495.

[22] Gehring, C., Coros, S., Hutter, M., Bloesch, M., A., H. M., and Siegwart, R., 2013. "Control of Dynamic Gaits for a Quadrupedal Robot". In IEEE Int. Conf. on Robotics and Automation.

[23] Hutter, M., and Gehring, C., 2012. MATLAB Tool proNEu Documentation. Tech. rep., ETH Zurich.

[24] Eigen C++ Library, Jan., 2014. http://eigen.tuxfamily.org.

[25] Boost C++ Libraries, Jan., 2014. http://www.boost.org.