

Complementarity Techniques for Minimal Coordinate Contact Dynamics

Harshavardhan Mylapilli

Department of Aerospace and
Mechanical Engineering,
University of Southern California,
Los Angeles, CA 90089
e-mail: mylapilli@usc.edu

Abhinandan Jain

Jet Propulsion Laboratory,
California Institute of Technology,
Pasadena, CA 91109
e-mail: Abhinandan.Jain@jpl.nasa.gov

In this paper, nonsmooth contact dynamics of articulated rigid multibody systems is formulated as a complementarity problem. Minimal coordinate (MC) formulation is used to derive the dynamic equations of motion as it provides significant computational cost benefits and leads to a smaller-sized complementarity problem when compared with the frequently used redundant coordinate (RC) formulation. Additionally, an operational space (OS) formulation is employed to take advantage of the low-order structure-based recursive algorithms that do not require mass matrix inversion, leading to a further reduction in these computational costs. Based on the accuracy with which Coulomb's friction cone is modeled, the complementarity problem can be posed either as a linear complementarity problem (LCP), where the friction cone is approximated using a polygon, or as a nonlinear complementarity problem (NCP), where the friction cone is modeled exactly. Both formulations are studied in this paper. These complementarity problems are further recast as nonsmooth unconstrained optimization problems, which are solved by employing a class of Levenberg–Marquardt (LM) algorithms. The necessary theory detailing these techniques is discussed and five solvers are implemented to solve contact dynamics problems. A simple test case of a sphere moving on a plane surface is used to validate these solvers for a single contact, whereas a 12-link complex pendulum example is chosen to compare the accuracy of the solvers for the case of multiple simultaneous contacts. The simulation results validate the MC-based NCP formulations developed in this paper. Moreover, we observe that the LCP solvers deliver accuracy comparable to that of the NCP solvers when the friction cone direction vectors in the contact tangent plane are aligned with the sliding contact velocity at each time step. The theory and simulation results show that the NCP approach can be seamlessly recast into an MC OS formulation, thus allowing for accurate modeling of frictional contacts, while at the same time reducing overall computational costs associated with contact and collision dynamics problems in articulated rigid body systems. [DOI: 10.1115/1.4033520]

1 Introduction

Over the past two decades, researchers have been developing complementarity-based formulations to solve contact and collision dynamics problems. Complementarity-based methods are an alternative to classical penalty-based methods which rely on a virtual spring–damper model to apply restoring forces at the point of deepest penetration between bodies in contact [1]. Penalty-based methods notoriously suffer from oscillatory effects and become numerically unstable when bodies collide with a high velocity. Small time steps and excessively damped implicit integrators are used to counter these problems, which makes the method slow and computationally expensive [2].

On the other hand, complementarity-based methods assume that the bodies are perfectly rigid and compute contact forces at each time step to prevent interpenetration. Complementarity methods use impulsive dynamics to handle collision and contact interactions. They avoid small time steps and numerical stiffening issues encountered with penalty methods by impulsively “stepping” over nonsmooth events [3]. There are two variants of the complementarity formulation: one variant uses an exact model of the friction cone which leads to an NCP, and the second variant employs a polyhedral approximation of the friction cone to yield a simplest LCP.

Considerable research effort [1,3,4] has been devoted into posing contact dynamics problems as solvable LCPs. The LCP method can however lead to inaccuracies as it relies on a discretized approximation of the friction cone. Increasing the accuracy of the LCP solution requires increasing the number of sides of the polygon used to approximate the friction cone, which leads to an increase in the number of ancillary variables in the problem. Increasing the number of these ancillary variables leads to a larger-sized LCP problem, and consequently, an increase in the computational cost. Moreover, the degree of alignment of the friction cone direction vectors in the contact tangent space with the tangential friction impulse has a significant effect on the accuracy of the solution [1]. In contrast, the NCP method does not require the use of direction vectors and has only three unknown variables per contact leading to a more compact formulation compared to the LCP approach [2].

Finding the solution to these complementarity problems is in general a nontrivial problem. Classical approaches to solving LCPs include pivoting methods such as Lemke's or Dantzig's algorithm [5], whereas iterative methods such as projected successive over-relaxation or projected Gauss–Seidel methods [6] are used to solve NCPs. More recent approaches (including the approach in this paper) recast these complementarity problems as nonsmooth unconstrained optimization problems, which are then solved using LM type of algorithms. This approach of reformulating the complementarity problem as an unconstrained optimization problem has been shown to perform exceedingly well [7–11].

For multilink systems, there are currently two main approaches to handle contact and collision dynamics problems: the classical *RC formulation* [4,12] and the *MC formulation* [3,13]. In the RC formulation, absolute coordinates are used to describe the motion of each link in the multilink robotic system. Each link in the

Contributed by the Design Engineering Division of ASME for publication in the *JOURNAL OF COMPUTATIONAL AND NONLINEAR DYNAMICS*. Manuscript received October 3, 2015; final manuscript received April 20, 2016; published online December 2, 2016. Assoc. Editor: Andreas Mueller.

The United States Government retains, and by accepting the article for publication, the publisher acknowledges that the United States Government retains, a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for United States government purposes.

system is treated as an independent body, and each interlink hinge is modeled explicitly as a bilateral constraint. On the other hand, in the MC formulation, the hinge (bilateral) constraints are automatically eliminated by choosing a minimal set of coordinates, which are used to describe the motion of the multilink robotic system. Any bilateral constraints that remain in the formulation arise only from loop-closure constraints, thereby leading to a smaller set of bilateral constraints when compared with the RC formulation. Note that in this paper, we focus on multibody systems with bilateral constraints and not on systems, such as granular material, which have no bilateral constraints and for whom there is no distinction between the RC and MC approaches.

In a recent paper, Jain [3] has demonstrated that the costs pertaining to these formulations can be categorized into two types—the cost of setting up the complementarity problem and the cost of solving the complementarity problem. Although, the complementarity problem for the RC formulation is easy to set up, it is computationally more expensive to solve since its size depends on the number of unilateral and bilateral constraints and the number of links in the system, which can be large. On the other hand, the MC formulation requires more work to set up the complementarity problem, but the size of the complementarity problem is much smaller since it depends only on the number of unilateral and bilateral constraints (and is independent of the number of links in the system). Thus, the reduction in the complementarity problem size shifts the computational burden from solving the complementarity problem to that of setting it up. By introducing certain low-order structure-based recursive algorithms (collectively referred to as OS algorithms) that do not require mass matrix inversion, Jain et al. [3,14] have shown that the MC approach leads to lower overall costs, with low costs for setting up as well as solving the complementarity problem. Because of these cost savings, when the MC OS formulation is applied to articulated rigid body systems, it leads to lower computational times when compared with the RC formulation [3]. Moreover, the constraint error management required for bilateral constraints needs to be enforced on a smaller set of constraints in the MC case as compared with the RC. Given these advantages and cost benefits, in this paper, we choose the MC OS formulation to study contact and collision dynamics problems in articulated rigid body systems.

The central focus of the present paper is to compare and contrast the linear and nonlinear complementarity approaches to solving contact dynamics problems in the context of MC OS formulation. Anitescu and Potra [4] and Trinkle et al. [1,12] formulate the contact dynamics problem as a mixed LCP (MLCP, a complementarity problem is termed *mixed* when both unilateral contact constraints and bilateral constraints are present in the dynamics formulation) but they use the RC formulation. Todorov et al. [2], on the other hand, formulate the contact dynamics problem as an MC NCP (although, as we will discuss later, they avoid having to solve an NCP through a suitable parameterization), but rely on expensive steps involving mass matrix inverses to set up the complementarity problem. Jain et al. [3,14] evaluate the MLCP approach in the framework of MC, but the MLCP is solved using the PATH solver [7] alone.

One of the contributions of this paper is to explore optimization-based approaches for solving the MC MLCP problem. In addition, we extend the MC OS MLCP formulation [3,14] by employing Todorov's approach [8], and consequently, eliminating the approximations associated with the discretization of the friction cone. In the process, we develop two linear and two nonlinear complementarity solvers, whose accuracy is analyzed for problems involving a single contact as well as multiple simultaneous contacts. Similar work comparing different linear and nonlinear complementarity solvers has been performed by Lacoursiere et al. [15] but they employ the RC formulation and a proximal function-based NCP solver [16].

Figure 1 depicts the various contact dynamics solver options studied in this paper. As discussed earlier, contact and collision dynamics problems are modeled using the MC OS formulation

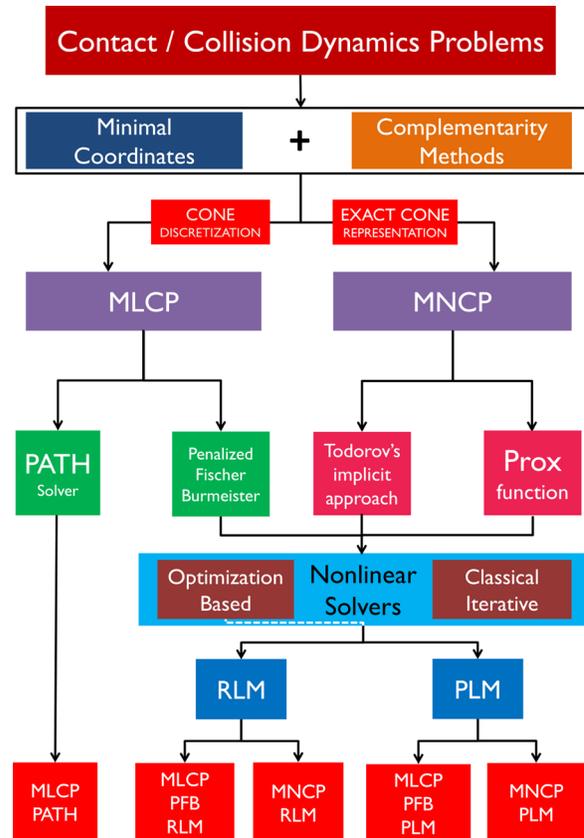


Fig. 1 Overview of the five contact dynamic solvers studied in this paper. Acronyms listed in the figure are defined in the Nomenclature section. The figure details the design choices that have been made while developing each of the five solvers. Contact dynamics problems are approached using the MC OS formulation and cast as a complementarity problem. Both linear and nonlinear complementarity formulations are studied. Among the many nonlinear complementarity formulations available in the literature, we consider Todorov's approach in the present study for its simplicity and ease of implementation, whereas the prox formulation, which is also widely used in the literature, is tabled for a future course of study.

and complementarity methods in the current study. Depending on the accuracy with which the friction cone is modeled, one ends up with an MLCP if the friction cone is approximated and a mixed NCP (denoted as MNCP) if the approximations are avoided. To solve the MLCP, the PATH solver [7] has been used in the literature [3]. Alternatively, one can make use of the penalized Fischer–Burmeister (PFB) function [17] to recast the MLCP as a set of nonlinear equations. On the other hand, of the many techniques available in the literature to solve an MNCP (Todorov's approach [8], proximal function-based approach [16], differential variational inequality approach [18], etc.), we focus our attention on Todorov's approach. A neat feature of Todorov's approach is that it reformulates the MNCP problem into an unconstrained (nonsmooth) optimization problem. Ultimately, all of the MNCP approaches lead to a system of nonlinear nonsmooth equations that need to be simultaneously solved. To numerically solve these nonlinear systems of equations, iterative solvers are commonly employed in the literature. These iterative solvers can be loosely classified as optimization-based solvers and nonoptimization-based solvers (for example, classical iterative solvers such as Gauss–Seidel and projected SOR methods). In the present study, we focus on optimization solvers and implement two variations of the LM algorithm [19] (the regular LM solver (RLM) [20] and the projected LM solver (PLM) [10]) to solve unconstrained

optimization problems. Thus, in summary, we develop five contact dynamics solvers in this study, namely, the MLCP–PATH solver, the MLCP–PFB–RLM solver, the MLCP–PFB–PLM solver, the MNCP–RLM solver, and the MNCP–PLM solver (see Nomenclature section for a description of these solvers).

This paper is organized as follows. In Sec. 2, we begin by introducing some fundamental concepts in contact and collision dynamics such as complementarity problems, constraints, dynamics formulations, Coulomb friction modeling, etc. In Sec. 3, MC formulation is used to formulate the dynamics as an MLCP by approximating the friction cone using a polyhedral approximation. This MLCP is further recast as an optimization problem by utilizing the PFB function. In Sec. 4, MC formulation is once again used along with an exact representation of the friction cone to formulate the dynamics as an MNCP and this is done by employing Todorov’s implicit approach. Furthermore, the optimization reformulation of the MNCP is also discussed. In Sec. 5, the unconstrained optimization algorithms are introduced and two variants of the LM-type of algorithms are discussed. In Sec. 6, the five contact dynamics solvers developed in this study are validated using the example of a sphere moving on a fixed horizontal plane, for which closed-form analytical solutions are available [1]. Subsequently, the example of a 12-link pendulum falling under gravity and colliding with its surrounding environment is used to compare and contrast the speed and accuracy of the five solvers. Finally in Sec. 7, we present our conclusions.

This paper is a revised and extended version of an earlier conference paper [21]. We have extended our previous work by adding a new section (Sec. 2.7) on the time evolution of nonsmooth dynamical systems. Next, in Sec. 2.2.2, we introduce the PFB function and contrast its performance against the regular Fischer–Burmeister (FB) function. Finally, the section on results and simulations has been rewritten with the addition of new figures and illustrations that provide a deeper insight into the advantages (and disadvantages) of the linear and nonlinear complementarity approaches to solving contact/collision dynamics problems.

2 Preliminaries

In this section, we review some fundamental concepts in contact and collision dynamics theory. The material has been adopted from Refs. [3,9,13,14,17,22–24]. We refer the interested reader to these references for a more in-depth handling of the subject matter presented herein.

2.1 Linear and Nonlinear Complementarity Problems.

The NCP [5] seeks a vector $z \in \mathbb{R}^n$ satisfying the following system of equations and inequalities:

$$\begin{aligned} z_i &\geq 0, & f_i(z) &\geq 0 \\ z_i \cdot f_i(z) &= 0 & \text{for } i = 1, 2, \dots, n \end{aligned} \quad (1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is any smooth nonlinear function. The *mixed nonlinear complementarity problem* (MNCP) is defined by the mapping $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, lower bounds $l_i \in \mathbb{R} \cup \{-\infty\}$, and upper bounds $u_i \in \mathbb{R} \cup \{+\infty\}$, where the solution of the MNCP is a vector $z \in \mathbb{R}^n$ such that for each $i \in \{1, 2, \dots, n\}$, one of the following alternatives holds:

$$\begin{aligned} f_i(z) &\geq 0 & \text{for } z_i &= l_i \\ f_i(z) &\leq 0 & \text{for } z_i &= u_i \\ f_i(z) &= 0 & \text{for } l_i < z_i < u_i \end{aligned} \quad (2)$$

The NCP and the MNCP problems reduce to the LCP and the MLCP problems, respectively, when f is an affine function of z , i.e.,

$$f(z) = \mathfrak{M}z + q \quad (3)$$

where \mathfrak{M} is an $n \times n$ matrix and q is an n -vector [5].

2.2 NCP Functions. To cast the nonlinear complementarity conditions (see Eq. (1)) as an equivalent nonlinear (nonsmooth) equation, *NCP functions* [25] (not to be confused with the nonlinear complementarity problem) are used in the literature, which have the property that

$$\phi(a, b) = 0 \iff a \geq 0, \quad b \geq 0, \quad ab = 0 \quad (4)$$

The equivalence condition guarantees that if a solution to the nonlinear equation ϕ is found, then it is also a solution to the NCP, assuming of course that the complementarity problem is solvable.

2.2.1 FB Function. Among the many *NCP functions* available in the literature (see Ref. [25] for a comparative study of popular *NCP functions*), the FB function [26] is probably the most widely used *NCP function* because of its desirable properties. The expression for the FB function is given by

$$\phi_{\text{FB}} := \phi_i(z_i, f_i(z)) = \sqrt{z_i^2 + f_i^2(z)} - z_i - f_i(z) \quad (5)$$

where $a := z_i$ and $b := f_i(z)$ in Eq. (4). The FB function has the property that the square of Eq. (5) is continuously differentiable [23]. This fact can be used to reformulate the set of nonlinear equations as an unconstrained optimization problem with a cost function $\psi : \mathbb{R}^n \rightarrow \mathbb{R}^+$ given by

$$\psi(z, f) = \frac{1}{2} \sum_{i=1}^n \phi_i^2(z_i, f_i) \quad (6)$$

Minimizing the cost function ψ gives us the solution to the system of nonlinear equations $\phi := \text{col}\{\phi_i(z_i, f_i(z))\} = 0$, which owing to Eq. (4) yields the solution to the complementarity problem.

To solve an MNCP, the *NCP function* ϕ_i takes the form [27]

$$\phi_i = \begin{cases} \sqrt{f_i^2(z) + z_i^2} - z_i - f_i(z) & \text{if } l_i = 0 \text{ and } u_i = \infty \\ -f_i(z) & \text{if } l_i = -\infty \text{ and } u_i = +\infty \end{cases} \quad (7)$$

where $l_i = 0$ and $u_i = \infty$ correspond to complementarity conditions, and $l_i = -\infty$ and $u_i = \infty$ correspond to equality conditions. The cost function for the mixed complementarity problem remains the same as Eq. (6).

2.2.2 PFB Function. One of the criticisms of the FB *NCP function* is that it is too flat in the positive orthant, which is the main region of interest for complementarity problems [17]. To comprehend this, consider a simple example with a single constraint: $n = 1$ and $f(z) = 0.1$ in Eq. (1) [17]. Clearly, $z^* = 0$ is a unique solution to this complementarity problem. However, suppose that a wild guess $z^{\text{guess}} = 10^{20}$ is chosen. When this guess is substituted into Eqs. (5) and (6), due to round off and cancellation errors, the cost function ψ takes a very small function value. This can trick the optimization process into wrongly assuming that it is close to the solution, when in reality z^{guess} is far away from z^* .

To overcome this drawback, Chen et al. [17] proposed the PFB function

$$\begin{aligned} \phi_{\text{PFB}} &:= \phi_i(z_i, f_i(z)) \\ &= \lambda \{-\phi_{\text{FB}}(z_i, f_i(z))\} + (1 - \lambda) \{\phi_+(z_i, f_i(z))\} \end{aligned} \quad (8)$$

where

$$\phi_+(a, b) = a_+ b_+ = \max(0, a) \cdot \max(0, b)$$

ϕ_{FB} is given by Eq. (5), and $0 < \lambda < 1$ is an arbitrary but fixed parameter. Thus, the PFB function ϕ_{PFB} is a convex combination of

ϕ_{FB} and ϕ_+ , where ϕ_+ penalizes the violations of the complementarity condition in the positive orthant. Note that the FB function contains a negative sign in the PFB formulation (see Eq. (8)) and the justification for this is presented in a lemma detailed in Ref. [27]. The PFB function reduces to the (negative of the) standard FB function for $\lambda = 1$.

To solve an MNCP, ϕ_i takes the form

$$\phi_i = \begin{cases} \left\{ \begin{array}{l} \lambda \{-\phi_{FB}(z_i, f_i(z))\} + \\ (1 - \lambda) \{\phi_+(z_i, f_i(z))\} \end{array} \right\} & \text{if } l_i = 0 \text{ and } u_i = \infty \\ -f_i(z) & \text{if } l_i = -\infty \text{ and } u_i = +\infty \end{cases} \quad (9)$$

The cost function ψ for these complementarity problems (i.e., Eqs. (8) and (9)) is, once again, given by Eq. (6). The results in this subsection hold true for LCPs and MLCPs as well. For both of these cases, the function $f(z)$ is an affine function of z (see Eq. (3)) as discussed earlier.

2.3 Constraints. In contact dynamics, the constraints between rigid links can be either *bilateral constraints* (for example, hinge constraints) defined by equality relationships of the form

$$b(x, \dot{x}, t) = 0 \quad (10)$$

or *unilateral contact constraints* that are defined by inequality relationships of the form

$$d(x, t) \geq 0 \quad (11)$$

where x denotes the vector of generalized coordinates of the system and t denotes time [13].

Equation (11) represents the nonpenetration condition between the surfaces of rigid bodies. The function $d(x, t)$ is referred to as the *distance* or *gap* function in the literature. Contact is said to occur when $d(x, t) = 0$. At the contact point, assuming sufficient smoothness, the surface normals are parallel for the bodies in contact. For a pair of bodies A and B in contact, the i th contact normal $\hat{n}(i)$ is defined as pointing from body B towards body A , such that the motion of A in the direction of the normal leads to a separation between the bodies [13]. A unilateral constraint is said to be *active* when there is contact, and the contact persists, i.e.,

$$d(x, t) = \dot{d}(x, t) = \ddot{d}(x, t) = 0 \quad (12)$$

The contact is said to be *inactive* when Eq. (12) is violated. Contact *separation* occurs when the relative linear velocity of the contact points along the normals becomes positive and the contact points drift apart. A separating constraint is in the process of losing contact and transitioning to an inactive state. At the start of a separation event, we have

$$d(x, t) = \dot{d}(x, t) = 0 \quad \text{and} \quad \ddot{d}(x, t) \geq 0 \quad (13)$$

Only active unilateral constraints generate constraint forces on the system [3].

2.4 Multibody Dynamics Formulations. In this subsection, we study three approaches for modeling the dynamics of multilink systems (that may possibly include closed-chain topologies). The three approaches illustrated here encompass the spectrum of modeling options available in the literature for analyzing the dynamics of multibody systems [28]. As mentioned earlier, the distinction between these approaches disappears for systems without any bilateral constraints and for those systems that contain a large number of independent bodies such as granular material, stack of

bricks, etc. In this paper, we are focusing on multilink systems with nonzero bilateral constraints.

2.4.1 RC Formulation. Classically, the dynamics of multibody systems are modeled using the *RC formulation* [1,13]. The RC formulation treats all bodies in the multilink system as independent (see Fig. 2) and has $6n$ degrees-of-freedom for an n -link system (the number of coordinates may be greater than $6n$ if coordinates such as quaternions are used for rotations). This system is further subject to unilateral contact constraints and explicit bilateral constraints. These bilateral constraints are associated with the interlink hinges which restrict the relative motion between the bodies.

The cost of setting up the complementarity problem in the RC formulation is of $O(n)$, and the size of the complementarity problem depends on the number of links n , the number of unilateral contact constraints, and the number of bilateral constraints. The advantages of this method include the relative ease with which the equations of motion can be set up, and the fact that the mass matrix of the system is block diagonal and constant, facilitating the use of sparse matrix solution techniques for solving the equations of motion. However, when articulated rigid body systems containing a large number of hinge constraints are considered, the large number of redundant coordinates present in the formulation, the need for constraint error management at each integration time step, and the use of differential-algebraic solvers work against the RC formulation.

2.4.2 MC Formulation. An alternative to the RC formulation is the *MC formulation* [3,13], where the interlink bilateral constraints are automatically eliminated by using a minimal set of coordinates that parameterize the permissible motion of the hinges (see Fig. 3). The number of coordinates associated with the hinge matches the number of degrees-of-freedom of the hinge. The system is therefore regarded as being composed of a tree-topology subsystem together with a minimal set of bilateral constraints arising from the remaining loop-closure constraints.

The advantage of the MC formulation is that the size of the complementarity problem is independent of the number of links n in the system and depends only on the number of unilateral and the number of bilateral constraints. Furthermore, the number of bilateral constraints in the MC case is much smaller (when compared with that of RC) as only loop-closure constraints enter the formulation. The underlying mathematical formulation is still of a differential-algebraic nature and constraint error management is still required, albeit for the smaller set of bilateral constraints.

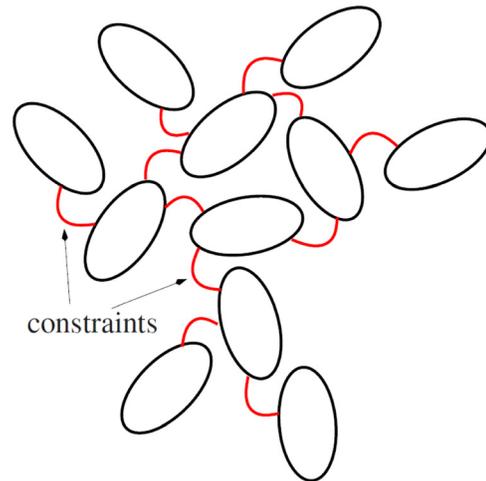


Fig. 2 RC formulation. Absolute coordinates are used to characterize each link in the system. Each interlink hinge is modeled explicitly as a bilateral constraint as illustrated. Figure adapted from Ref. [3].

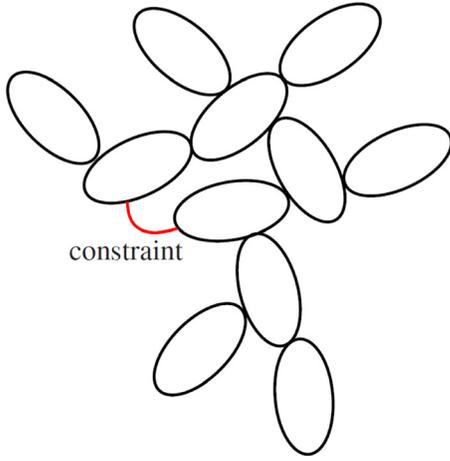


Fig. 3 MC formulation. A minimal set of coordinates are used to characterize the dynamics of the system, which is obtained by eliminating all the interlink bilateral constraints. Only loop-closure constraints remain in the formulation. Figure adapted from Ref. [3].

The reduction in size of the complementarity problem shifts the burden from solving the complementarity problem to setting up the complementarity problem [3]. To set up the complementarity problem, the mass matrix of the system needs to be inverted, and given that the mass matrix in the MC case is dense and configuration dependent, the computational cost of its inverse can be considerable. However, by taking advantage of low-order spatial operator algorithms (called as OS algorithms) that do not require mass matrix inversion to solve for the system dynamics, the set up cost of the complementarity problem can be reduced to $O(n)$ complexity, where n here refers to the number of minimal coordinates.

The benefits of the MC OS approach come to the fore when considering articulated rigid body systems. Jain [3] demonstrated that the size of the MLCP as well as the time taken to solve the MLCP increases proportional to the number of links in the system for the RC MLCP approach, whereas the size of the MLCP in the MC OS (MC-OS MLCP) approach remains the same regardless of the number of links in the system. Thus, the MC approach along with the OS formulation results in lower overall costs for setting up the complementarity problem and for solving it. We take a closer look at the approach in Sec. 2.5.

2.4.3 Constraint Embedding (CE) Formulation. Although not the focus of the current paper, for the sake of completeness, we briefly introduce a third formulation, referred to as the CE formulation [22,28] for modeling dynamics of multibody systems that include closed-chain topologies. The CE formulation uses the MC approach as its starting point and eliminates loop-closure bilateral constraints that are left over in the MC formulation by aggregating the bodies affected by the closure constraint as compound bodies (see Fig. 4).

The system is therefore effectively a tree topology with neither the hinge constraints nor the loop-closure constraints entering the formulation. The CE formulation requires additional steps to set up compared to the MC formulation. However, unlike the MC formulation, the nature of the underlying mathematical formulation of the CE method is that of ordinary differential equations (instead of differential-algebraic equations like in the MC case), and constraint error management techniques are not required as bilateral constraints do not enter the formulation. The preservation of the tree-topology facilitates the use of structure-based tree algorithms that translate into faster computational times for the CE formulation when compared with the MC formulation [28].

2.5 MC Formulation: A Closer Look. Let \mathcal{N} denote the number of degrees-of-freedom of the tree subsystem. The

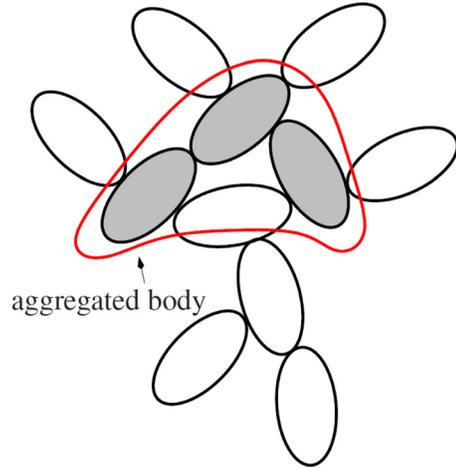


Fig. 4 CE formulation. This formulation is an extension to the MC formulation, where not only the interlink hinge constraints but also the loop-closure constraints are eliminated. Bodies affected by eliminating the closure constraints are aggregated and modeled as compound bodies. Figure adapted from Ref. [3].

equations of motion [3] for the tree-topology subsystem are given by

$$\mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) = \tau \quad (14)$$

where $\theta \in \mathbb{R}^{\mathcal{N}}$ is the vector of hinge coordinates, $\mathcal{M}(\theta) \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ is the configuration-dependent, symmetric, and positive-definite inertia matrix, $\mathcal{C}(\theta, \dot{\theta}) \in \mathbb{R}^{\mathcal{N}}$ is the vector of Coriolis, gyroscopic, and gravitational forces acting on the system, and $\tau \in \mathbb{R}^{\mathcal{N}}$ denotes the vector of applied generalized forces.

2.5.1 Bilateral Constraints. Let n_b denote the dimension of bilateral constraints arising from loop closures in the system. Since n_b in the MC approach corresponds only to the loop-closure constraints, this number is much smaller than the n_b in the RC approach. There exists a full-rank matrix $G_b(\theta, t) \in \mathbb{R}^{n_b \times \mathcal{N}}$ and a vector $\mathbf{U}(t) \in \mathbb{R}^{n_b}$ that defines the velocity domain constraint equation, which can be expressed as

$$G_b(\theta, t)\dot{\theta} = \mathbf{U}(t) \quad (15)$$

The bilateral constraints effectively reduce the independent degrees-of-freedom of the system from \mathcal{N} to $\mathcal{N} - n_b$. The smooth dynamics of closed-chain systems can be obtained by modifying the tree system dynamics in Eq. (14) to include the effect of the bilateral constraints via Lagrange multipliers, $\lambda \in \mathbb{R}^{n_b}$, as follows:

$$\begin{aligned} \mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) - G_b^T(\theta, t)\lambda &= \tau \\ G_b(\theta, t)\dot{\theta} &= \mathbf{U}(t) \end{aligned} \quad (16)$$

where $-G_b^T(\theta, t)\lambda$ term in the first equation of Eq. (16) represents the internal constraint forces arising from the loop-closure constraints.

2.5.2 Unilateral Constraints. We now introduce unilateral contact constraints in this formulation. Let n_u denote the number of unilateral contact nodes and $\nu_u \in \mathbb{R}^{3n_u}$ denote the vector of relative linear velocities across the contact nodes. The mapping between the contact velocities ν_u and the body spatial velocities θ is defined by a matrix $G_u \in \mathbb{R}^{3n_u \times \mathcal{N}}$ such that

$$\nu_u = G_u \dot{\theta} \quad (17)$$

The matrix G_u also maps the impulses at the contact node pairs, $F_u \in \mathbb{R}^{3n_u}$, to the corresponding generalized impulses, $p_u \in \mathbb{R}^{\mathcal{N}}$, by means of the following dual mapping:

$$p_u = G_u^T F_u \quad (18)$$

The smooth dynamics equations of motion in Eq. (16) can be extended to include the effect of these contact impulses p_u as follows:

$$\begin{bmatrix} \mathcal{M} & -G_b^T \\ G_b & 0 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \lambda \end{bmatrix} = \begin{bmatrix} (\tau - C) + p_u / \Delta_t \\ \bar{\mathbf{U}} \end{bmatrix} \quad (19)$$

where $\bar{\mathbf{U}} = \mathbf{U} \cdot (t) - \dot{G}_b \dot{\theta} \in \mathbb{R}^{n_b}$ and Δ_t is the time step. The conversion of impulses into forces above assumes the use of first-order time integration, which we will discuss in Sec. 2.7. Note that to solve for the equations of motion, we need to determine the unknown impulses, F_u , at the contact node pairs.

2.6 Contact Impulses and Coulomb Friction Modeling. To describe the rolling and sliding phenomena at the i th active contact constraint node, the three-dimensional contact impulse vector $F_u(i) \in \mathbb{R}^3$ and contact velocity vector $\nu_u(i) \in \mathbb{R}^3$ can be decomposed into normal and tangential components as [3]

$$\begin{aligned} F_u(i) &= F_n(i) \hat{\eta}(i) + F_t^T(i) \hat{t}(i) \\ &= F_n(i) \hat{\eta}(i) + [F_f(i) \quad F_o(i)] \begin{bmatrix} \hat{f}(i) \\ \hat{o}(i) \end{bmatrix} \\ &= F_n(i) \hat{\eta}(i) + F_f(i) \hat{f}(i) + F_o(i) \hat{o}(i) \end{aligned} \quad (20)$$

$$\begin{aligned} \nu_u(i) &= \nu_n(i) \hat{\eta}(i) + \nu_t^T(i) \hat{t}(i) \\ &= \nu_n(i) \hat{\eta}(i) + [\nu_f(i) \quad \nu_o(i)] \begin{bmatrix} \hat{f}(i) \\ \hat{o}(i) \end{bmatrix} \\ &= \nu_n(i) \hat{\eta}(i) + \nu_f(i) \hat{f}(i) + \nu_o(i) \hat{o}(i) \end{aligned} \quad (21)$$

where $\hat{\eta}(i)$ is the contact normal; $\hat{t}(i)$ is the tangent plane vector in the contact tangent plane, which is further spanned by two orthogonal vectors $\hat{f}(i)$ and $\hat{o}(i)$. $F_n(i) \in \mathbb{R}$ represents the normal component of the contact impulse, and $F_t(i) = [F_f(i) \quad F_o(i)]^T \in \mathbb{R}^2$ is the tangential (friction) component of the contact impulse (where $F_f(i)$ and $F_o(i)$ are components of $F_t(i)$ along the $\hat{f}(i)$ and $\hat{o}(i)$ directions, respectively). Similarly, $\nu_n(i) \in \mathbb{R}$ and $\nu_t(i) = [\nu_f(i) \quad \nu_o(i)]^T \in \mathbb{R}^2$ represent the normal and tangential components of the linear relative velocity of the body at the i th contact pair. Specifically, $\nu_n(i)$ denotes the relative velocity which is normal to the contact point, and $\nu_t(i)$ represents the relative velocities that are unconstrained but are resisted by friction.

As discussed earlier, an active i th contact is defined by $\mathfrak{d}(i) = 0$ (the bodies are touching) and $\nu_n(i) = 0$. Moreover, the contact is said to be *sliding* [12] when

$$\nu_n(i) = 0 \quad \text{and} \quad \nu_t(i) \neq 0 \quad (22)$$

On the other hand, the contact is said to be *rolling* [12] when

$$\nu_n(i) = \nu_t(i) = 0 \quad (23)$$

Having defined the concepts of rolling and sliding, we can now state Coulomb's law of friction [8] as

$$\begin{aligned} F_n(i) &\geq 0, \quad \nu_n(i) \geq 0, \quad F_n(i) \nu_n(i) = 0 \\ \nu_t(i) &\text{ parallel to } F_t(i), \quad \langle \nu_t(i), F_t(i) \rangle \leq 0 \\ \|F_t(i)\| &\leq \mu(i) F_n(i) \end{aligned} \quad (24)$$

The first line of Eq. (24) states that the normal force and the normal contact velocity cannot both be simultaneously positive. The normal force is zero when the bodies are separating and positive

when there is sustained contact. The second line of Eq. (24) implies that if there is sliding between the bodies in contact, then the tangential friction impulse should lie in a direction that is opposite to the tangential relative linear velocity. This statement is also referred to as the *principle of maximum dissipation*. The third line states that the tangential friction impulse must lie inside the friction cone. The tangential friction impulse is on the boundary of the cone when the bodies are sliding and in the interior of the cone when the bodies are rolling. The coefficient of friction is denoted by μ . Notice that only the first line of Eq. (24) is a strict complementarity condition whereas additional work needs to be done to bring the other two conditions into the complementarity framework.

2.7 Time Evolution of Nonsmooth Dynamical Systems.

Integration schemes for time evolution of nonsmooth mechanical systems can be classified into two categories: *event-driven* schemes and *time-stepping* schemes [29]. Event-driven schemes separate motion into piecewise smooth and nonsmooth intervals. These schemes are fairly accurate, but are not well suited when there are frequent transitions between the intervals in a short amount of time [16]. Time-stepping schemes, on the other hand, need no such separation of motion into smooth and nonsmooth intervals and handle impacts and impact-free motion within the same time step. Infinite switching sequences are often handled within one single time increment [24]. Although these schemes are only first-order accurate, they are extensively used in the literature because of their robustness and ease of implementation.

There are many different time-stepping schemes that are available in the literature for handling time integration of nonsmooth systems (see Refs. [24] and [29]). Here, we review two of the most widely used schemes: semi-implicit Euler's scheme [1,3] and Moreau's midpoint rule [24,30]. The superscripts +, m, and - denote the value of the mathematical quantity at the start, middle, and end of the time step, respectively.

2.7.1 Semi-Implicit Euler's Scheme. The semi-implicit Euler's scheme [1,3] is a first-order symplectic time-stepping scheme. For a step size Δ_t , let $\theta^- = \theta(t)$ denote the value of θ at the start of the time step, and $\theta^+ = \theta(t + \Delta_t)$ denote the value of θ at the end of the time step. A similar notation is followed for defining the generalized velocities $\dot{\theta}^+$ and $\dot{\theta}^-$. The quantities $\mathcal{M}^- = \mathcal{M}(\theta^-)$, $C^- = C(\theta^-, \dot{\theta}^-)$, τ , G_u , and G_b are all calculated at the beginning of the time step and are assumed to be constant over the entire time step. With these quantities known, our goal is to derive approximations for the generalized velocities, $\dot{\theta}^+$, at the end of the time step. This is given by [3]

$$\begin{aligned} \dot{\theta}^+ &= \dot{\theta}^- + \ddot{\theta} \Delta_t \\ &= \dot{\theta}^- + \{\mathcal{M}^-\}^{-1} (\tau - C^- + G_u^T F_u + G_b^T \lambda) \Delta_t \end{aligned} \quad (25)$$

At any given time step, a check is made to find out if there are any bodies in contact or are interpenetrating. If there are no bodies in contact or are interpenetrating, then the contact forces, F_u , are zero and we have smooth motion. On the other hand, if there are bodies in contact or are interpenetrating, then the calculations are reverted to the start of the time step, a complementarity problem is solved to find the contact forces, which are then used to propagate the velocities forward (see Eq. (25)). These contact forces are assumed to be constant over the entire step size, and thus, small time steps are required for accurate results. With an estimate for $\dot{\theta}^+$ at our disposal, the positions are propagated as follows [3]:

$$\theta^+ = \theta^- + \dot{\theta}^+ \Delta_t \quad (26)$$

2.7.2 Moreau's Midpoint Rule. Moreau's midpoint rule [24,30] is extensively used in the literature in conjunction with the

proximal function method. But the time-stepping scheme can be modified, as shown below, for use with complementarity formulations similar to the ones described in this paper.

Unlike semi-implicit Euler's scheme where the states at the end of the time step are estimated using values at the beginning of the step, Moreau's scheme first estimates the midpoint positions as

$$\theta^m = \theta^- + \frac{1}{2} \Delta_t \dot{\theta}^- \quad (27)$$

and then estimates the positions and velocities at the end of the time step (i.e., θ^+ and $\dot{\theta}^+$) by using values of the estimated midpoint positions (θ^m), and the velocities at the beginning of the time step ($\dot{\theta}^-$) as shown below

$$\begin{aligned} \dot{\theta}^+ &= \dot{\theta}^- + \ddot{\theta} \Delta_t \\ &= \dot{\theta}^- + \{\mathcal{M}^m\}^{-1} (\tau - \mathcal{C}^{m,-} + \mathbf{G}_u^T F_u + \mathbf{G}_b^T \lambda) \Delta_t \end{aligned} \quad (28)$$

and

$$\theta^+ = \theta^- + \frac{\dot{\theta}^- + \dot{\theta}^+}{2} \Delta_t \quad (29)$$

where the quantities $\mathcal{M}^m = \mathcal{M}(\theta^m)$, $\mathcal{C}^{m,-} = \mathcal{C}(\theta^m, \dot{\theta}^-)$, τ , \mathbf{G}_u , and \mathbf{G}_b are all computed using θ^m and $\dot{\theta}^-$. The unknown quantities that remain in Eqs. (28) and (29) are θ^+ , $\dot{\theta}^+$, and F_u .

To make the scheme explicit, one can first compute the contact forces (as in the semi-implicit Euler's scheme) by solving the complementarity problem. The only difference between the Euler's scheme and this explicit method is that the complementarity problem is solved with the mass matrix and other requisite quantities evaluated using the midpoint position estimates and the velocities at the beginning of the time step (instead of using both positions and velocities at the beginning of the time step as in semi-implicit Euler's scheme). The rest of the procedure remains the same as the Euler's scheme with the exception that Eq. (29) is used to compute the positions instead of Eq. (26).

On a separate note, the implicit counterparts of Euler's scheme and Moreau's schemes typically allow for the use of larger time steps for the same level of accuracy as compared to the explicit ones. To make these schemes implicit, consider Eq. (25) for Euler's case (or Eq. (28) for Moreau's case). These equations represent a set of nonlinear equations with unknowns $\dot{\theta}^+$ and F_u that can be iteratively solved to find the unknown quantities [30]. This iterative procedure can be more expensive since we may need to solve multiple complementarity problems to arrive at the solution for a single time step. The positions are, once again, propagated using Eq. (26) for Euler's case (and Eq. (29) for Moreau's case).

The final form of the MLCP (see Eq. (43)) and the MNCP (see Eq. (55)) depends on the specific choice of the state propagation scheme that is employed (see Eqs. (37) and (54) where the time discretization is introduced into the MLCP and MNCP formulations, respectively). In the present paper, we use the semi-implicit Euler's scheme to develop these expressions, though it should be straightforward to adapt this procedure to alternate state propagation schemes (such as Moreau's midpoint rule and Moreau-Jean schemes).

3 Discretized Friction Cone MLCP Formulation

In this section, we summarize the MC MLCP formulation for contact and collision dynamics [3,14]. Coulomb's friction conditions (described in Sec. 2.6) are inherently nonlinear. However, one can linearize these conditions by approximating the friction cone using a pyramid [1]. This allows us to formulate the contact dynamics problem as a mixed linear complementarity problem (MLCP).

For a unilateral contact, Fig. 5 depicts a circular friction limit set (denoted by the circle) of radius μF_n in the contact tangent

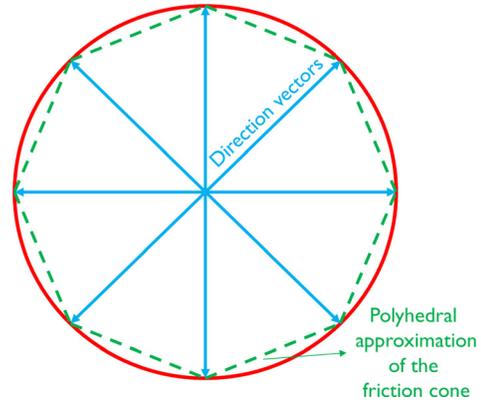


Fig. 5 Approximating the friction cone by a friction polyhedron using a finite number of direction vectors. The polygon is spanned by the direction vectors as illustrated in the figure. For each direction vector in the set, we include the opposite direction vector as well, so that the entire friction cone is spanned by the direction vectors (as opposed to a sector of the cone). Figure adapted from Ref. [3].

plane. The circular set is approximated by a convex polygon (illustrated by the dashed lines), which is spanned with the help of direction vectors (as illustrated by the solid arrows). The number of direction vectors and their orientations in the contact tangent plane can be arbitrary. However, by choosing these direction vectors and their orientations (as discussed later on in this section), one can help mitigate some of the errors associated with the discretization of the friction cone.

3.1 Friction Cone Discretization. Consider that the friction cone at the i th contact is approximated by a friction polyhedron consisting of a finite number n_f of unit direction vectors $\hat{d}_j(i)$ in the tangent plane. For notational simplicity, we assume that n_f is the same across all the contact points. The tangential friction impulse for the i th contact is expressed as the linear combination of these direction vectors as [3]

$$F_t(i) \hat{i}(i) = \sum_{j=1}^{n_f} \beta_j(i) \hat{d}_j(i) = \mathbf{D}(i) \beta(i) \quad (30)$$

where

$$\mathbf{D}(i) = [\hat{d}_1(i), \hat{d}_2(i), \dots, \hat{d}_{n_f}(i)] \in \mathbb{R}^{3 \times n_f}$$

and

$$\beta(i) = \text{col}\{\beta_j(i)\}_{j=1}^{n_f} \in \mathbb{R}^{n_f}$$

Combining Eqs. (20) and (30), we have

$$F_u(i) = \mathbf{D}(i) \underline{\beta}(i) \quad (31)$$

where

$$\underline{\mathbf{D}}(i) = [\hat{n}(i), \mathbf{D}(i)] \in \mathbb{R}^{3 \times (n_f+1)}$$

and

$$\underline{\beta}(i) = \begin{bmatrix} F_n(i) \\ \beta(i) \end{bmatrix} \in \mathbb{R}^{n_f+1}$$

During sliding, the $\beta_j(i)$ component is nonzero and is equal to $\mu(i) F_n(i)$ for just the single direction j that corresponds to the

closest direction opposing the tangential relative linear velocity. Denoting $\sigma(i) = \|\nu_t(i)\|$

$$\beta_k(i) = \begin{cases} \mu(i)F_n(i), & \text{if } \sigma(i) > 0 \text{ and } k = j \\ 0, & \text{if } \sigma(i) > 0 \text{ and } k \neq j \end{cases}$$

The sliding and rolling contact relationships of Eq. (24) can now be rephrased as the following complementarity conditions [3]:

$$\begin{aligned} & \hat{\eta}^T(i)\nu_u^+(i) \perp F_n(i) \\ & \sigma(i)E(i) + D^T(i)\nu_u^+(i) \perp \beta(i) \\ & \mu(i)F_n(i) - E^T(i)\beta(i) \perp \sigma(i) \end{aligned} \quad (32)$$

where $E(i) = \text{col}\{1\}_{j=1}^{n_r} \in \mathbb{R}^{n_r}$ and the component of the relative linear velocity along the contact normal is $\nu_n^+(i) = \hat{\eta}^T(i)\nu_u^+(i)$, where the superscript $+$ ($-$) denotes the value of the quantity just after (before) the application of an impulse. Using Eq. (31), the above complementarity conditions can be more compactly expressed as

$$\begin{aligned} & \hat{E}(i)\sigma(i) + \underline{D}^T(i)\nu_u^+(i) \perp \underline{\beta}(i) \\ & \bar{E}(i)\underline{\beta}(i) \perp \sigma(i) \end{aligned} \quad (33)$$

where

$$\hat{E}(i) = \begin{bmatrix} 0 \\ E(i) \end{bmatrix} \in \mathbb{R}^{n_r+1}$$

and

$$\bar{E}(i) = [\mu(i), -E^T(i)] \in \mathbb{R}^{1 \times (n_r+1)}$$

At the system level, these conditions across all the contacts can be expressed as [14]

$$\begin{aligned} & \hat{E}\sigma + \underline{D}^T\nu_u^+ \perp \underline{\beta} \\ & \bar{E}\underline{\beta} \perp \sigma \end{aligned} \quad (34)$$

where

$$\begin{aligned} \underline{\beta} &= \text{col}\{\beta(i)\}_{i=1}^{n_u} \in \mathbb{R}^{n_u(n_r+1)} \\ \sigma &= \text{col}\{\sigma(i)\}_{i=1}^{n_u} \in \mathbb{R}^{n_u} \\ \underline{D} &= \text{diag}\{D(i)\}_{i=1}^{n_u} \in \mathbb{R}^{3n_u \times n_u(n_r+1)} \\ \hat{E} &= \text{diag}\{\hat{E}(i)\}_{i=1}^{n_u} \in \mathbb{R}^{n_u(n_r+1) \times n_u} \\ \bar{E} &= \text{diag}\{\bar{E}(i)\}_{i=1}^{n_u} \in \mathbb{R}^{n_u \times n_u(n_r+1)} \end{aligned}$$

and

$$\nu_u^+ = \text{col}\{\nu_u^+(i)\}_{i=1}^{n_u} \in \mathbb{R}^{3n_u}$$

Furthermore, the contact impulses at the system level can be written as [3]

$$F_u = \underline{D}\underline{\beta} \quad (35)$$

where

$$F_u = \text{col}\{F_u(i)\}_{i=1}^{n_u} \in \mathbb{R}^{3n_u}$$

In principle, there are no restrictions on the number of direction vectors that can be chosen or their orientation in the contact tangential plane. Reducing the number of direction vectors reduces the size of the LCP problem but increases the approximation error. However, when choosing direction vectors, one can help reduce the approximation errors for the MLCP case by

- (a) aligning *one* of the direction vectors in our set at each time step to coincide with the opposite of the tangential relative linear velocity vector at that time step (i.e., $-\nu_t^-$). This ensures that at least one of the direction vectors lines up closely with tangential friction impulse (F_t^+), thereby reducing approximation errors (see Eq. (30)).
- (b) including the opposite direction vector in the set for each direction vector chosen as this helps span the entire friction cone. As a consequence of this, *one* of the direction vectors is also close in direction to the tangential sliding velocity (i.e., ν_t^+) (see Eq. (32)).

It should be noted, however, that these measures only mitigate the approximation errors arising from the misalignment of direction vectors and the tangential friction impulse, but do not entirely eliminate them. This is one of the main drawbacks of the linear complementarity approach. We discuss these issues more rigorously in Sec. 6.

3.2 Setting Up the MLCP. We now set up the OS MLCP formulation [3,13]. Equation (19) can be rearranged as

$$\begin{aligned} & G_b\mathcal{M}^{-1}[\tau - \mathcal{C} + G_b^T\lambda + G_u^T\underline{D}\underline{\beta}/\Delta_t] - \bar{\mathbf{U}} = 0 \\ & G_b\mathcal{M}^{-1}G_b^T p_b + G_b\mathcal{M}^{-1}G_u^T \underline{D}\underline{\beta} + \alpha_b^f \Delta_t = 0 \end{aligned} \quad (36)$$

where

$$p_b = \lambda\Delta_t$$

and

$$\alpha_b^f = G_b\mathcal{M}^{-1}(\tau - \mathcal{C}) - \bar{\mathbf{U}}$$

The relative linear acceleration of the contact nodes, $\dot{\nu}_u$, is obtained by differentiating Eq. (17) with respect to time. Employing the semi-implicit Euler's scheme (see Sec. 2.7.1), ν_u^+ can be computed using $\dot{\nu}_u$ as follows:

$$\begin{aligned} \nu_u^+ &= \nu_u^- + \dot{\nu}_u\Delta_t = \nu_u^- + (G_u\ddot{\theta} + \dot{G}_u\dot{\theta})\Delta_t \\ &= \nu_u^- + G_u\mathcal{M}^{-1}G_b^T p_b + G_u\mathcal{M}^{-1}G_u^T \underline{D}\underline{\beta} + \alpha_u^f \Delta_t \end{aligned} \quad (37)$$

where

$$\alpha_u^f = G_u\mathcal{M}^{-1}(\tau - \mathcal{C}) + \dot{G}_u\dot{\theta}$$

Denoting n_c as the number of the combined set of nodes associated with the unilateral and bilateral constraints of the system, the spatial velocities of these nodes are given by the stacked vector $\mathcal{V}_c \in \mathbb{R}^{6n_c}$, which is related to $\dot{\theta}$ by

$$\mathcal{V}_c = \mathcal{J}\dot{\theta} \quad (38)$$

where $\mathcal{J} \in \mathbb{R}^{6n_c \times \mathcal{N}}$ is the Jacobian of the constraint nodes. Now, there exist matrices $Q_u \in \mathbb{R}^{3n_u \times 6n_c}$ and $Q_b \in \mathbb{R}^{3n_b \times 6n_c}$ such that

$$\nu_u = Q_u\mathcal{V}_c = Q_u\mathcal{J}\dot{\theta} \quad (39)$$

and

$$\mathfrak{U} = Q_b\mathcal{V}_c = Q_b\mathcal{J}\dot{\theta} \quad (40)$$

Comparing Eq. (39) with Eq. (17) and Eq. (40) with Eq. (15), we obtain

$$G_u = Q_u\mathcal{J} \quad (41)$$

and

$$G_b = Q_b \mathcal{J} \quad (42)$$

Denoting $\underline{\Lambda} = \mathcal{J} \mathcal{M}^{-1} \mathcal{J}^T \in \mathbb{R}^{6n_c \times 6n_c}$, Eqs. (34)–(37) can be expressed as [3]

$$\begin{aligned} Q_b \underline{\Lambda} Q_b^T p_b + Q_b \underline{\Lambda} Q_u^T \underline{D} \underline{\beta} + \alpha_b^f \Delta_t &= 0 \\ \underline{D}^T Q_u \underline{\Lambda} Q_b^T p_b + \underline{D}^T Q_u \underline{\Lambda} Q_u^T \underline{D} \underline{\beta} + \hat{E} \sigma + \underline{D}^T (\alpha_u^f \Delta_t + \nu_u^-) \perp \underline{\beta} \\ \bar{E} \underline{\beta} \perp \sigma \end{aligned} \quad (43)$$

or more compactly as

$$f(z) = \mathfrak{M}z + q \quad (44)$$

where

$$\begin{aligned} \mathfrak{M} &= \begin{bmatrix} X \underline{\Lambda} X^T & E_1 \\ E_2 & \mathbf{0} \end{bmatrix} \\ z &= \begin{bmatrix} p_b \\ \underline{\beta} \\ \sigma \end{bmatrix} \\ q &= \begin{bmatrix} \alpha_b^f \Delta_t \\ \underline{D}^T (\alpha_u^f \Delta_t + \nu_u^-) \\ \mathbf{0} \end{bmatrix} \\ X &= \begin{bmatrix} Q_b \\ \underline{D}^T Q_u \end{bmatrix} \\ E_1 &= \begin{bmatrix} \mathbf{0} \\ \hat{E} \end{bmatrix} \\ E_2 &= [\mathbf{0} \quad \bar{E}] \end{aligned}$$

Equation (44) is a $(n_b + n_u(n_f + 2))$ sized MLCP where the first equation is an equality condition while the bottom two equations are linear complementarity conditions. Structure-based OS recursive algorithms of order $O(N) + O(n_c^2)$ can be used to compute the configuration dependent matrix $\underline{\Lambda}$ as shown in Refs. [3,13,22].

Previous investigators employed the PATH solver [7] to solve the MLCP (described by Eq. (44)). Jain [3] has implemented such a solver in his study, which we refer to as the MLCP–PATH solver in this paper. Part of the focus of this paper is to explore alternate methods to solve the MLCP besides using the PATH solver (see Sec. 3.3 where the FB and PFB functions are used to cast the MLCP as an unconstrained optimization problem).

3.3 Casting the MLCP as an Optimization Problem. As discussed in Sec. 2.2, the mixed linear complementarity conditions of Eq. (44) can be reformulated as a system of nonlinear nonsmooth equations using *NCP functions* [10,23,25]. The resulting nonlinear equations can be further recast as an unconstrained minimization problem. This allows us to employ the multitude of optimization solvers available for solving such problems. The optimization solvers require a cost function (and its gradient) to find the minimum point. Depending on the specific choice of the *NCP function*, the expressions for these quantities vary, and in what follows below, we provide these quantities for the FB function and its penalized counterpart.

3.3.1 FB Function. For the MLCP described by Eq. (44), the cost vector ($\phi = \text{col}\{\phi_i\}$) for the FB formulation is calculated

using Eq. (7), its cost function ψ using Eq. (6), and its gradient by $\nabla \psi = \phi^T J$, where J is the Jacobian matrix, whose entries $J_{ij} = [\partial \phi_i / \partial z_j]$ are given by

(a) If $l_i = 0$ and $u_i = \infty$, then

$$\frac{\partial \phi_i}{\partial z_i} = \begin{cases} z_i + f_i \mathfrak{M}_{ii} - 1 - \mathfrak{M}_{ii}, & \text{if } \|z_i, f_i(z)\| \neq 0 \\ \sqrt{1 + \mathfrak{M}_{ii}^2} - 1 - \mathfrak{M}_{ii}, & \text{if } \|z_i, f_i(z)\| = 0 \end{cases}$$

and

$$\frac{\partial \phi_i}{\partial z_j} = \begin{cases} \frac{f_i \mathfrak{M}_{ij}}{\|z_i, f_i(z)\|} - \mathfrak{M}_{ij}, & \text{if } \|z_i, f_i(z)\| \neq 0 \\ 0, & \text{if } \|z_i, f_i(z)\| = 0 \end{cases}$$

(b) If $l_i = -\infty$ and $u_i = \infty$, then $(\partial \phi_i / \partial z_j) = -\mathfrak{M}_{ij}$.

3.3.2 PFB Function. On the other hand, for the PFB function formulation, the cost vector ϕ for the MLCP (described by Eq. (44)) is computed using Eq. (9), the cost function ψ using Eq. (6), and its gradient by $\nabla \psi = \phi^T J$, where the procedure illustrated in Ref. [17] (and reproduced here for convenience) is adopted to compute the Jacobian matrix J .

(a) If $l_i = 0$ and $u_i = \infty$, then formulate sets $S_1 = \{i \mid z_i = f_i(z) = 0\}$ and $S_2 = \{i \mid z_i > 0, f_i(z) > 0\}$. Construct an n -vector y such that $y_i = 0$ for $i \notin S_1$, and $y_i = 1$ for $i \in S_1$. Let $J_i = [(\partial \phi_i / \partial z_1) \quad \dots \quad (\partial \phi_i / \partial z_j) \quad \dots \quad (\partial \phi_i / \partial z_n)]$ denote the i th row of the Jacobian matrix J and e_i denote the i th row of the identity matrix.

Then, if $i \in S_1$

$$\begin{aligned} J_i &= \lambda \left(1 - \frac{y_i}{\|y_i, \nabla f_i(z)^T y\|} \right) e_i^T \\ &\quad + \lambda \left(1 - \frac{\nabla f_i(z)^T y}{\|y_i, \nabla f_i(z)^T y\|} \right) \nabla f_i(z)^T \end{aligned} \quad (45)$$

Next, if $i \in S_2$

$$\begin{aligned} J_i &= \left[\lambda \left(1 - \frac{z_i}{\|z_i, f_i(z)\|} \right) + (1 - \lambda) f_i(z) \right] e_i^T \\ &\quad + \left[\lambda \left(1 - \frac{f_i(z)}{\|z_i, f_i(z)\|} \right) + (1 - \lambda) z_i \right] \nabla f_i(z)^T \end{aligned} \quad (46)$$

Alternatively, if $i \notin S_1 \cup S_2$

$$\begin{aligned} J_i &= \lambda \left(1 - \frac{z_i}{\|z_i, f_i(z)\|} \right) e_i^T \\ &\quad + \lambda \left(1 - \frac{f_i(z)}{\|z_i, f_i(z)\|} \right) \nabla f_i(z)^T \end{aligned} \quad (47)$$

(b) If $l_i = -\infty$ and $u_i = \infty$, then

$$\begin{aligned} J_i &= \begin{bmatrix} \partial \phi_i / \partial z_1 & \dots & \partial \phi_i / \partial z_j & \dots & \partial \phi_i / \partial z_n \end{bmatrix} \\ &= [-\mathfrak{M}_{i1} \quad \dots \quad -\mathfrak{M}_{ij} \quad \dots \quad -\mathfrak{M}_{in}] \end{aligned} \quad (48)$$

Note that for $\lambda = 1$, the formulation above reduces to (the negative of) the standard FB function formulation.

In our investigations, we found that for certain values of λ , the PFB function performs better (converges to the solution in lesser

number of iterations) whereas for others, it performs worse than the standard FB function. The optimal value of this parameter λ is dependent on the problem at hand. When performing contact dynamics simulations over a time interval, multiple (different) complementarity problems need to be evaluated over a series of time steps. To choose an optimal value of λ for all of these steps can be challenging and the choice is often made through trial and error (as has been done in this paper). However, when the value of λ is appropriately chosen, the PFB function takes considerably lesser number of iterations to converge to the (respective) solutions at each time step (and over the duration of the simulation) when compared with the standard FB function.

With this optimization reformulation (using either the standard FB function or the PFB function), unconstrained minimization algorithms (see Sec. 5) can be employed to compute the minimum point, and hence the solution to the MLCP problem.

4 Exact Friction Cone MNCP Formulation

We now turn our attention to a nonlinear complementarity formulation of contact dynamics that avoids the friction cone approximations required by the LCP formulation. Our objective is to use the low-cost MC OS formulation and avoid the friction cone approximations by recasting the dynamics as an MNCP. The MNCP formulation, unlike the MLCP formulation, models the friction conditions exactly.

4.1 Exact Modeling of Friction Cone. Until now, we have only dealt with an approximate model of the friction cone. However, to model the friction cone exactly, an alternative approach is required that captures Coulomb's friction conditions (see Eq. (24)) exactly. Of the many approaches available in the literature that accurately capture Coulomb's phenomena [8,16,18], we employ Todorov's implicit approach [2,8]. The appealing features of Todorov's approach include its simplicity and ease of implementation coupled with the fact that the need to solve an MNCP is bypassed by reformulating the MNCP as a set of nonlinear equations through a suitable parameterization. A method similar to Todorov's approach has also been postulated by Drumwright and Shell [31].

Realizing that the contact relative linear velocity $\nu_u(i)$ and the contact impulse $F_u(i)$ at the i th active contact constraint node are not independent but are instead coupled through laws of contact and friction, Todorov [8] parameterizes both $F_u(i)$ and $\nu_u(i)$ using an *unconstrained* nonphysical variable $\mathfrak{z}(i)$ given by

$$\mathfrak{z}(i) = \begin{bmatrix} \mathfrak{z}_r(i) \\ \mathfrak{z}_o(i) \\ \mathfrak{z}_n(i) \end{bmatrix} \in \mathfrak{R}^3$$

The normal force $F_n(i)$ and normal velocity $\nu_n(i)$ are encoded by $\mathfrak{z}_n(i)$ as [8]

$$\begin{aligned} F_n(i) &= \max(0, -\mathfrak{z}_n(i)) \\ \nu_n(i) &= \max(0, \mathfrak{z}_n(i)) \end{aligned} \quad (49)$$

On the other hand, the tangential force

$$F_t(i) = \begin{bmatrix} F_r(i) \\ F_o(i) \end{bmatrix} \in \mathfrak{R}^2$$

and the tangential velocity

$$\nu_t(i) = \begin{bmatrix} \nu_r(i) \\ \nu_o(i) \end{bmatrix} \in \mathfrak{R}^2$$

are encoded by the tangential components of $\mathfrak{z}(i)$, i.e.,

$$\mathfrak{z}_t(i) = \begin{bmatrix} \mathfrak{z}_r(i) \\ \mathfrak{z}_o(i) \end{bmatrix} \in \mathfrak{R}^2$$

as

$$\begin{aligned} F_t(i) &= -s(i)\mathfrak{z}_t(i) \\ \nu_t(i) &= [1 - s(i)]\mathfrak{z}_t(i) \end{aligned} \quad (50)$$

where

$$s(i) = \min\left(1, \frac{\mu(i)F_n(i)}{\|\mathfrak{z}_t(i)\|}\right)$$

The role of these parameters can be understood as follows:

- (1) **Nonpenetration condition:** The nonpenetration condition requires one of $F_n(i)$ or $\nu_n(i)$ to be positive, while the other to be zero (see first line of Eq. (24) and its equivalent parametric representation in Eq. (49)). A pictorial representation of Eq. (49) is shown in Fig. 6. When $\mathfrak{z}_n(i) > 0$, the i th contact constraint is inactive, and $\mathfrak{z}_n(i)$ represents the non-zero relative normal linear velocity $\nu_n(i) = \mathfrak{z}_n(i)$, with the normal force $F_n(i) = 0$. Consequently, $F_t(i) = [0, 0]^T$ because there is no contact. On the other hand, when $\mathfrak{z}_n(i) < 0$, the contact constraint is active, and $\mathfrak{z}_n(i)$ represents the nonzero normal force $F_n(i) = -\mathfrak{z}_n(i)$ with the normal velocity $\nu_n(i) = 0$. Thus, $\mathfrak{z}_n(i)$ characterizes the nonpenetration condition and the complementarity relation between the variables $F_n(i)$ and $\nu_n(i)$.
- (2) **Rolling contact:** When the i th contact is active (i.e., $\mathfrak{z}_n(i) < 0$, $F_n(i) = -\mathfrak{z}_n(i)$ and $\nu_n(i) = 0$), and $s(i) = 1$ (i.e., when $\|\mathfrak{z}_t(i)\| \leq \mu(i)F_n(i)$), the contact is said to be *rolling*. Since $s(i) = 1$ for a rolling contact, $\nu_t(i) = [0, 0]^T$ and $F_t(i) = -\mathfrak{z}_t(i)$ in accordance with Eq. (50). Furthermore, since $\|\mathfrak{z}_t(i)\| = \|\mathfrak{z}_t(i)\| = \|\mathfrak{z}_t(i)\| \leq \mu(i)F_n(i)$, we stay inside the circular friction limit circle as depicted in Fig. 7. Thus, the rolling conditions described in Sec. 2.6 have been parametrically represented in terms of the unconstrained variable $\mathfrak{z}(i)$.
- (3) **Sliding contact:** When the i th contact is active (i.e., $\mathfrak{z}_n(i) < 0$, $F_n(i) = -\mathfrak{z}_n(i)$ and $\nu_n(i) = 0$), and $s(i) < 1$ (i.e., when $\|\mathfrak{z}_t(i)\| > \mu(i)F_n(i)$), the contact is said to be *sliding*. Since $0 < s(i) < 1$ for a sliding contact, the tangential contact velocity $\nu_t(i) = (1 - s(i))\mathfrak{z}_t(i)$ and the tangential friction impulse $F_t(i) = -s(i)\mathfrak{z}_t(i)$ always lie in opposite directions as required. Furthermore, $\|F_t(i)\| = \mu(i)F_n(i)$, which means that we are on the boundary of the friction cone as depicted in Fig. 8. The remainder of the $\mathfrak{z}_t(i)$ vector is interpreted as the sliding velocity $\nu_t(i)$. Thus, the sliding

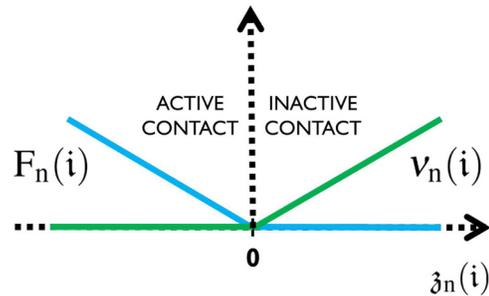


Fig. 6 Nonpenetration conditions. The unconstrained variable $\mathfrak{z}_n(i)$ captures the complementarity relationship between the normal friction impulse $F_n(i)$ and the normal contact velocity $\nu_n(i)$. Figure adapted from Ref. [8].

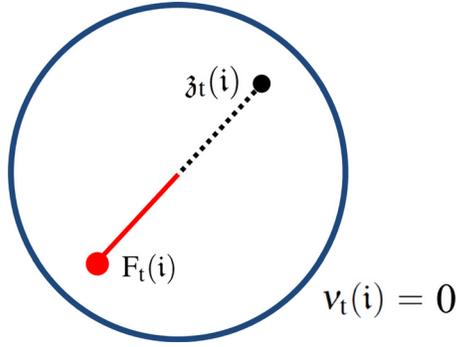


Fig. 7 Rolling conditions. The dotted line denotes $\mathfrak{z}_t(i)$, the solid line denotes the tangential friction impulse $F_t(i)$, whereas the circle represents the circular friction limit set. For an (active) rolling contact, the tangential friction impulse (whose direction is opposite to that of $\mathfrak{z}_t(i)$) lies inside the friction cone, and the tangential contact velocity $\nu_t(i)$ is zero. The rolling conditions can thus be parameterized in terms of $\mathfrak{z}(i)$. Figure adapted from Ref. [8].

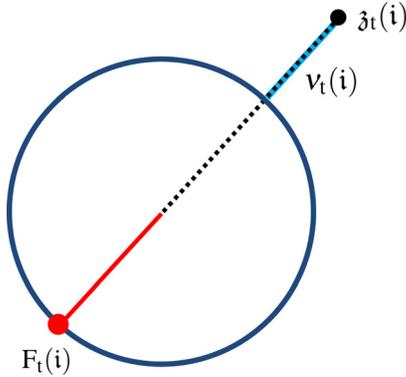


Fig. 8 Sliding conditions. The dotted line denotes $\mathfrak{z}_t(i)$, the (dark) solid line denotes the tangential friction impulse vector $F_t(i)$, the (light) solid line denotes the tangential contact velocity vector $\nu_t(i)$, whereas the circle represents the circular friction limit set. For an (active) sliding contact, $F_t(i)$ and $\nu_t(i)$ should lie in opposite directions. The vector $\mathfrak{z}_t(i)$ is partitioned into $F_t(i)$ and $\nu_t(i)$. When the contact is sliding, $F_t(i)$ lies on the boundary of the friction cone, whereas the remainder of the vector $\mathfrak{z}_t(i)$ is interpreted as the sliding velocity $\nu_t(i)$. The sliding conditions can thus be parameterized in terms of $\mathfrak{z}(i)$. Figure adapted from Ref. [8].

conditions discussed in Sec. 2.6 have also been parametrically formulated in terms of the unconstrained variable $\mathfrak{z}(i)$.

For an in-depth discussion of Todorov's approach, the interested reader is referred to Ref. [8]. The set of equations – Eqs. (49) and (50) – which model Coulomb's friction conditions exactly can be more compactly expressed as [8]

$$\nu_u(i) = F_u(i) + \mathfrak{z}(i) \quad (51)$$

In summary, Todorov's implicit approach involves designing functions $F_u(i)$ and $\nu_u(i)$ (as described by Eqs. (49) and (50)) in terms of an unconstrained nonphysical variable $\mathfrak{z}(i)$ such that Coulomb's friction conditions (given by Eq. (24)), which are inherently nonlinear, are precisely captured for any $\mathfrak{z}(i)$. Through this parameterization via $F_u(i)$ and $\nu_u(i)$, one is essentially left with a system of nonlinear nonsmooth equations in terms of unconstrained $\mathfrak{z}(i)$, which can be solved by employing the multitude of nonlinear optimization routines available in the literature. The procedure to do this is illustrated in Secs 4.2, 4.3, and 5.

4.2 Setting Up the MNCP. We now develop the MC OS MNCP formulation using Todorov's approach. Using the system level contact impulse, Eq. (19) can be rearranged as

$$G_b \mathcal{M}^{-1} G_b^T p_b + G_b \mathcal{M}^{-1} G_u^T F_u + \alpha_b^f \Delta_t = 0 \quad (52)$$

which can be further simplified using $G_u = Q_u J$ and $G_b = Q_b J$ as

$$Q_b \underline{\Delta} Q_b^T p_b + Q_b \underline{\Delta} Q_u^T F_u + \alpha_b^f \Delta_t = 0 \quad (53)$$

Employing the semi-implicit Euler's scheme (see Sec. 2.7.1), the relative linear velocity of the contact nodes post application of the contact impulse (i.e., ν_u^+) is given by

$$\begin{aligned} \nu_u^+ &= \nu_u^- + \dot{\nu}_u \Delta_t = \nu_u^- + (G_u \ddot{\theta} + \dot{G}_u \dot{\theta}) \Delta_t \\ &= \nu_u^- + G_u \mathcal{M}^{-1} G_b^T p_b + G_u \mathcal{M}^{-1} G_u^T F_u + \alpha_u^f \Delta_t \\ &= \nu_u^- + Q_u \underline{\Delta} Q_b^T p_b + Q_u \underline{\Delta} Q_u^T F_u + \alpha_u^f \Delta_t \end{aligned} \quad (54)$$

Using Eq. (51), Eqs. (53) and (54) can be rewritten as

$$\begin{aligned} \Phi_b &= Q_b \underline{\Delta} Q_b^T p_b + Q_b \underline{\Delta} Q_u^T F_u(\mathfrak{z}) + \alpha_b^f \Delta_t = 0 \\ \Phi_u &= Q_u \underline{\Delta} Q_b^T p_b + (Q_u \underline{\Delta} Q_u^T - I) F_u(\mathfrak{z}) - \mathfrak{z} + \alpha_u^f \Delta_t + \nu_u^- = 0 \end{aligned} \quad (55)$$

which can be expressed in matrix form as

$$\begin{aligned} \Phi(p_b, \mathfrak{z}) &= \begin{bmatrix} \Phi_b \\ \Phi_u \end{bmatrix} \\ &= \begin{bmatrix} Q_b \underline{\Delta} Q_b^T & Q_b \underline{\Delta} Q_u^T \\ Q_u \underline{\Delta} Q_b^T & (Q_u \underline{\Delta} Q_u^T - I) \end{bmatrix} \begin{bmatrix} p_b \\ F_u(\mathfrak{z}) \end{bmatrix} \\ &\quad + \begin{bmatrix} \alpha_b^f \Delta_t \\ \alpha_u^f \Delta_t + \nu_u^- - \mathfrak{z} \end{bmatrix} \end{aligned} \quad (56)$$

where Φ is called the residual and I is the identity matrix. As mentioned earlier, the computational cost for evaluating the configuration-dependent $\underline{\Delta}$ matrix in the above expressions can be significantly reduced by using structure-based OS recursive algorithms [3,13,22].

4.3 Optimization Reformulation of the MNCP. The problem now reduces to solving a set of nonlinear nonsmooth equations $\Phi=0$, which can be reformulated as an unconstrained minimization problem with the cost function given by

$$\psi = \frac{1}{2} \Phi^T \Phi \quad (57)$$

The gradient of the cost is $\nabla \psi = \Phi^T J$ where the Jacobian J can be computed as

$$\begin{aligned} J &= \begin{bmatrix} \frac{\partial \Phi_b}{\partial p_b} & \frac{\partial \Phi_b}{\partial \mathfrak{z}} \\ \frac{\partial \Phi_u}{\partial p_b} & \frac{\partial \Phi_u}{\partial \mathfrak{z}} \end{bmatrix} \\ &= \begin{bmatrix} Q_b \underline{\Delta} Q_b^T & Q_b \underline{\Delta} Q_u^T \\ Q_u \underline{\Delta} Q_b^T & (Q_u \underline{\Delta} Q_u^T - I) \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \frac{\partial F_u}{\partial \mathfrak{z}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -I \end{bmatrix} \end{aligned} \quad (58)$$

where

$$\frac{\partial F_u}{\partial \mathfrak{z}} = \text{diag} \left(\frac{\partial F_u(1)}{\partial \mathfrak{z}(1)}, \frac{\partial F_u(2)}{\partial \mathfrak{z}(2)}, \dots, \frac{\partial F_u(n_u)}{\partial \mathfrak{z}(n_u)} \right)$$

and n_u is the number of unilateral contacts at the current integration time step. Focusing on a single contact (and dropping the index) for convenience, the expression for an individual 3×3 block matrix is given by

$$\frac{\partial F_u}{\partial \mathfrak{z}} = \frac{\partial}{\partial \mathfrak{z}} \begin{bmatrix} F_f \\ F_o \\ F_n \end{bmatrix} = \begin{bmatrix} \frac{\partial F_f}{\partial \mathfrak{z}_f} & \frac{\partial F_f}{\partial \mathfrak{z}_o} & \frac{\partial F_f}{\partial \mathfrak{z}_n} \\ \frac{\partial F_o}{\partial \mathfrak{z}_f} & \frac{\partial F_o}{\partial \mathfrak{z}_o} & \frac{\partial F_o}{\partial \mathfrak{z}_n} \\ \frac{\partial F_n}{\partial \mathfrak{z}_f} & \frac{\partial F_n}{\partial \mathfrak{z}_o} & \frac{\partial F_n}{\partial \mathfrak{z}_n} \end{bmatrix} \quad (59)$$

where the entries of the matrix $\partial F_u / \partial \mathfrak{z}$ are as follows:

$$\begin{aligned} \frac{\partial F_f}{\partial \mathfrak{z}_f} &= \begin{cases} -s & \text{if } \frac{\mu F_n}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} \geq 1 \\ \frac{-s \mathfrak{z}_o^2}{\|\mathfrak{z}_f, \mathfrak{z}_o\|^2} & \text{if } \frac{\mu F_n}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} < 1 \end{cases} \\ \frac{\partial F_f}{\partial \mathfrak{z}_o} &= \begin{cases} 0 & \text{if } \frac{\mu F_n}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} \geq 1 \\ \frac{s \mathfrak{z}_o \mathfrak{z}_f}{\|\mathfrak{z}_f, \mathfrak{z}_o\|^2} & \text{if } \frac{\mu F_n}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} < 1 \end{cases} \\ \frac{\partial F_f}{\partial \mathfrak{z}_n} &= \begin{cases} 0 & \text{if } \frac{\mu F_n}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} \geq 1 \\ \frac{-\mu}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} \frac{\partial F_n}{\partial \mathfrak{z}_n} & \text{if } \frac{\mu F_n}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} < 1 \end{cases} \\ \frac{\partial F_o}{\partial \mathfrak{z}_f} &= \begin{cases} 0 & \text{if } \frac{\mu F_n}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} \geq 1 \\ \frac{s \mathfrak{z}_f \mathfrak{z}_o}{\|\mathfrak{z}_f, \mathfrak{z}_o\|^2} & \text{if } \frac{\mu F_n}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} < 1 \end{cases} \\ \frac{\partial F_o}{\partial \mathfrak{z}_o} &= \begin{cases} -s & \text{if } \frac{\mu F_n}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} \geq 1 \\ \frac{-s \mathfrak{z}_f^2}{\|\mathfrak{z}_f, \mathfrak{z}_o\|^2} & \text{if } \frac{\mu F_n}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} < 1 \end{cases} \\ \frac{\partial F_o}{\partial \mathfrak{z}_n} &= \begin{cases} 0 & \text{if } \frac{\mu F_n}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} \geq 1 \\ \frac{-\mu}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} \frac{\partial F_n}{\partial \mathfrak{z}_n} & \text{if } \frac{\mu F_n}{\|\mathfrak{z}_f, \mathfrak{z}_o\|} < 1 \end{cases} \\ \frac{\partial F_n}{\partial \mathfrak{z}_f} &= 0, \quad \frac{\partial F_n}{\partial \mathfrak{z}_o} = 0 \\ \frac{\partial F_n}{\partial \mathfrak{z}_n} &= \begin{cases} -1 & \text{if } \mathfrak{z}_n < 0 \\ 0 & \text{if } \mathfrak{z}_n \geq 0 \end{cases} \end{aligned}$$

Such optimization problems are usually handled using some form of Gauss–Newton methods, e.g., the LM algorithms [19] which we consider in more detail in Sec. 5.

5 Unconstrained Optimization Algorithms

In this section, we discuss some methods for solving unconstrained optimization problems of the form [32]: minimize $\psi(x)$, where $\psi: \mathbb{R}^n \rightarrow \mathbb{R}^+$ is convex and twice continuously differentiable. We assume that the optimization problem is solvable; in other words, there exists an optimal value at x^* denoted by p^* such that $p^* = \psi(x^*)$ is the minimum. If ψ is differentiable and convex, a necessary and sufficient condition for x^* to be optimal is that the gradient of ψ at x^* is zero, i.e., $\nabla \psi(x^*) = 0$ which yields n equations in n variables, the solution to which gives us the optimal value of $\psi(x)$ [32]. The most practical methods to solve convex minimization problems involve iterative algorithms where a sequence of points $x_1, x_2, \dots, x_k \in \text{dom } \psi$ are produced with $\psi(x_k) \rightarrow p^*$ as $k \rightarrow \infty$. The algorithm terminates when $\psi(x_k) - p^* \leq \varepsilon$, where ε is some specified tolerance very close to zero. Extensions to nonsmooth problems, where the function ψ is continuous but may not be differentiable at finitely many points, are discussed in Refs. [8] and [9].

LM Algorithm. Newton-like methods often fail to work properly as the Hessian computed at each iteration step can become singular during the optimization process, and this consequently stalls the algorithm. Moreover, in some cases, the Newton step being computed may not be a step in the descent direction leading to erroneous results. The LM algorithm [19], also known as the damped least squares algorithm, has been proposed in the literature as a safeguard against these issues, and has the added benefit that it converges to the minimum irrespective of the starting position [20,32].

The LM algorithm interpolates between Newton's method and the method of gradient descent. Gradient methods are guaranteed to reach the minimum point although they sometimes take a long time to converge, whereas Newton's method is known for its speed of convergence to the minimum. By combining both methods, the advantages of each of these methods are combined. The LM algorithm is guaranteed to reach the minimum, and it does so taking a small number of steps. Furthermore, the LM algorithm is more robust compared to Newton's method; as in many cases, we do end up finding the minimum point even for poor initial guesses.

One disadvantage of the LM algorithm is that the (approximate) Hessian needs to be computed, stored, and inverted, which deteriorates the performance of the algorithm when the Hessian is large in size. Regardless, the LM algorithm is widely used by researchers attempting to solve complementarity problems (recast as optimization problems) [8–10]. Below we discuss two such LM algorithms—the regular LM algorithm (RLM) and the projected LM algorithm (PLM).

5.1 RLM Algorithm. The RLM algorithm [20] switches adaptively between the gradient descent method and Newton's method based on a reduction factor r . The reduction factor r is defined as the ratio of the actual decrease of the function ψ and its quadratic approximation q at $x = x_k$. The closer the reduction factor r is to unity, the more reliable the quadratic approximation is, and the smaller we can allow λ_{LM} to be. Thus, if r is small, λ_{LM} is increased, and if r is large, λ_{LM} is reduced. If λ_{LM} is close to zero, then the RLM becomes a Newton's method. Alternately, if λ_{LM} is large, then this makes the Hessian diagonally dominant and we end up with a gradient descent method. Typical parameters that work well empirically with this method are $r_{\min} = 0.3$, $r_{\max} = 0.8$, $\lambda_{LM} = 10^{-3}$, and $\lambda_{\text{step}} = 20$. The following algorithmic implementation of RLM has been adopted from Bazaraa et al. [20].

ALGORITHM: Regular LM algorithm (RLM)

given
 $\psi(x) = \frac{1}{2} \phi(x)^T \phi(x)$, initial $x \in \text{dom } \psi$, tol $\varepsilon = 10^{-30} > 0$,
 $\lambda_{LM} = 10^{-3}$, $\lambda_{\text{step}} = 20$, $\lambda_{\min} = 10^{-25}$
 repeat

- (1) Compute the following quantities at the k^{th} iterate x_k :
 - Cost $\psi(x_k)$,
 - Jacobian $J(x_k) = (\partial\phi/\partial x)$,
 - (Approximate) Hessian $H(x_k) = J^T J + \lambda_{\text{LM}} I$,
 - Step size $\Delta x_k = H^{-1}(J^T \phi)$
- (2) Stopping Criteria: If $\psi(x_k) < \varepsilon$, then STOP
- (3) Update $x_{k+1} := x_k + \Delta x_k$
- (4) Compute the Reduction Factor r

$$r = \frac{\psi(x_{k+1}) - \psi(x_k)}{q(x_{k+1}) - q(x_k)}$$
 where $q(x_k)$ is the quadratic approximation of $\psi(x_k)$
- (5) Adaptively change λ based on the reduction r :
 - if $r < 0.3$, then $\lambda_{\text{LM}} := \lambda_{\text{LM}} * \lambda_{\text{step}}$
 - if $r > 0$, then accept Δx and if $r > 0.8$, then $\lambda_{\text{LM}} := \lambda_{\text{LM}} / \lambda_{\text{step}}$

5.2 Projected LM Algorithm. The main difference between the projected LM algorithm (PLM) [10] and the RLM algorithm is in the computation of the search direction. In the PLM algorithm, the search direction is computed as $\Delta x_k = H^{-1}B$, where $H_{2n \times n}$ is generated by stacking the $n \times n$ Jacobian matrix J with the $n \times n$ diagonal matrix whose (diagonal) entries are given by $\sqrt{\lambda_{\text{LM}}}$ (where λ_{LM} is the LM damping parameter). Similarly, the $2n \times 1$ vector B is generated by stacking the vector $(-\phi)$ with a zero vector of size n . The following basic implementation of the PLM algorithm has been adopted from Refs. [10] and [11].

ALGORITHM: Projected LM algorithm (PLM)

given
 $\psi(x) = \frac{1}{2} \phi(x)^T \phi(x)$, initial $x \in \text{dom} \psi$, tol $\varepsilon = 10^{-30}$
 > 0 , $\lambda_{\text{LM}} = 10^{-16}$
 repeat

- (1) Compute the following quantities at the k^{th} iterate x_k :
 - Cost $\psi(x_k)$,
 - Jacobian $J(x_k) = \partial\phi/\partial x$,
 - appended-Jacobian $H(x_k) = \begin{bmatrix} J \\ \sqrt{\lambda_{\text{LM}}} I \end{bmatrix}$,
 - appended- ϕ $B(x_k) = \begin{bmatrix} -\phi(x_k) \\ 0 \end{bmatrix}$,
 - Step size $\Delta x_k = H^{-1}B$

- (2) Stopping Criteria: If $\psi(x_k) < \varepsilon$, then STOP
- (3) Update $x_{k+1} := x_k + t\Delta x_k$

To this basic algorithm, additional features can be added such as the nonmonotone line search technique [33] together with watchdog stabilization [34] to make the algorithm more robust. For example, if the best function value found so far has not been sufficiently reduced within a fixed number of iterations, then the program is restarted from that point using a monotone line search [10].

We make use of these RLM and PLM optimization routines to develop four schemes that solve contact dynamics problems. The first two schemes, i.e., MLCP–PFB–RLM and the MLCP–PFB–PLM, make use of the MLCP–PFB optimization reformulation (see Sec. 3.3), and the optimization problem is solved using the RLM and PLM algorithms, respectively. The last two schemes, i.e., MNCP–RLM and the MNCP–PLM solvers, use Todorov’s formulation (see Sec. 4.3), and the optimization problem is solved, once again, using RLM and PLM algorithms, respectively. We validate these routines and test the speed and accuracy of these schemes in Sec. 6.

6 Simulation Results

To validate the accuracy of the contact dynamics solvers developed in this study, we consider the following examples:

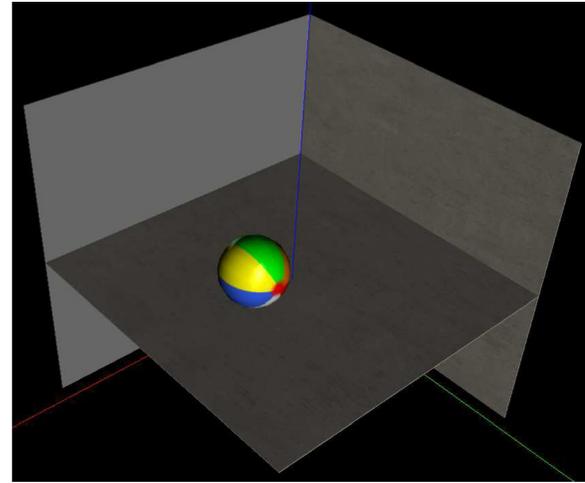


Fig. 9 Uniform sphere moving on a horizontal plane surface

- (1) Uniform sphere sliding and rolling on a fixed horizontal plane: This example has been adopted from Ref. [1] where an exact analytical solution has been derived for a specific set of initial conditions. The example involves a single

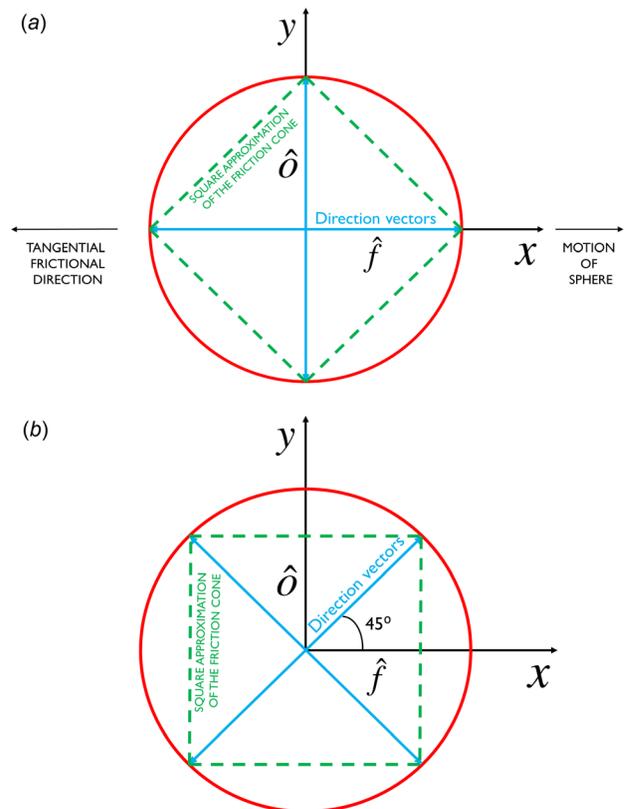


Fig. 10 Polyhedral approximation of the friction cone for the MLCP set of solvers: (a) case 1 (validation case)—one of the friction cone direction vectors is aligned precisely with the tangential friction impulse direction and (b) case 2—friction cone direction vectors are misaligned with tangential friction impulse direction. The circle represents the circular friction limit set, the solid arrows denote the direction vectors, and the dotted lines denote the polygonal approximation of the circular friction limit set. The motion of the sphere is along the positive x -direction (and so is the tangential contact velocity vector), whereas the tangential friction impulse vector is along the negative x -direction. Four direction vectors that are at right angles to each other are chosen to span the friction cone.

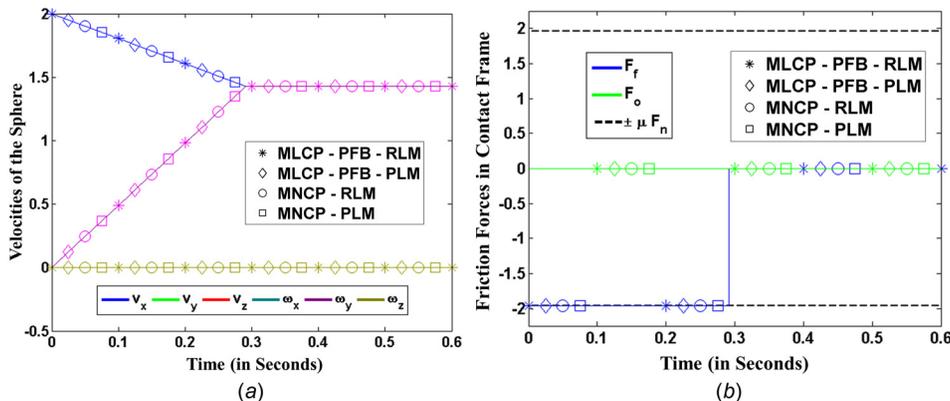


Fig. 11 Motion data for the sphere example (case 1), where the friction cone direction vectors are aligned with the tangential contact velocity: (a) time history of the generalized velocities and (b) time history of friction forces in contact coordinates

contact (that of the sphere in contact with the horizontal plane) at each time step. The analytical solution from Ref. [1] is used to validate the solvers developed in this paper.

- (2) Twelve-link complex pendulum colliding with itself and its surrounding environment: This example of a 12-link swinging pendulum is used to evaluate the accuracy of the linear and nonlinear complementarity solvers developed in this paper for the case of multiple simultaneous contacts.

6.1 Sphere on a Fixed Horizontal Plane. Consider a uniform sphere placed on a fixed horizontal plane in the presence of a uniform gravitational field as shown in Fig. 9. The mass and radius of the sphere are unity, whereas the initial configuration and initial velocity of the center of mass of the sphere are specified by

$$\theta_o = [001 | 1000]$$

and

$$\hat{\theta}_o = [v_{x_o} \ v_{y_o} \ v_{z_o} | \omega_{x_o} \ \omega_{y_o} \ \omega_{z_o}] = [200 | 000]$$

respectively. The first three numbers of θ_o specify the location of the center of mass of the sphere, whereas the last four numbers specify the initial orientation of frame attached to the center of mass (represented via a unit quaternion). On the other hand, $\hat{\theta}_o$ specifies that the sphere is given an initial velocity of 2 m/s in the x -direction. The coefficient of friction is assumed to have a constant value of

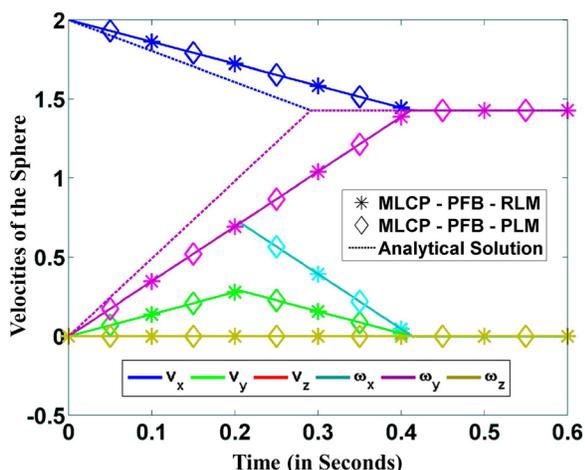


Fig. 12 Time history of generalized velocities for the sphere example (case 2), where the friction cone direction vectors are *not* aligned with the tangential contact velocity

$\mu = 0.2$. The frictional \hat{f} , orthogonal \hat{o} , and normal \hat{n} directions of the contact frame have been chosen to lie exactly along the x -, y -, and z -directions of the inertial frame, respectively (see Fig. 10(a)).

According to the analytical solution [1], for the initial conditions and parameters described earlier, the sphere is supposed to slide for a duration of 0.291 s in the x -direction and then roll indefinitely thereafter. The numerical solutions are computed for a duration of $t = 0.6$ s with a time step $\Delta t = 1$ ms. We consider two cases for the sphere example. In case 1, we compare the numerical solutions computed using the MC MNCP (and the MC MLCP) solvers with the analytically derived solution. On a separate note, the MLCP set of solvers rely on the polyhedral approximation of the friction cone, and the choice of the number of direction vectors and their orientation at each time step can have significant impact on the accuracy of the solution. To illustrate this impact, we consider two variations of the polyhedral friction cone approximation (see cases 1 and 2 and Fig. 10) for the MLCP solvers in this example.

Case 1: We choose four direction vectors (i.e., $n_f = 4$ in Eq. (30)) at right angles to each other in the contact tangent plane to discretize the friction cone (Fig. 10(a)). One of the direction vectors is chosen to be aligned perfectly with the tangential velocity v_t^- , and its opposite vector is also included. We believe that such an alignment is the best choice for orienting the direction vectors of the MLCP solvers since it reduces the representation errors for the tangential contact velocity as well as the tangential contact impulse.

Case 2: The direction vectors for Case 2 are similar to Case 1 with the exception that vectors are rotated by a 45 degree angle such that none of the vectors align with or oppose v_t^- (Fig. 10(b)).

The MNCP set of solvers do not use direction vectors and are thus unaffected by the choices of these direction vectors. They are considered only in Case 1 for validation purposes.

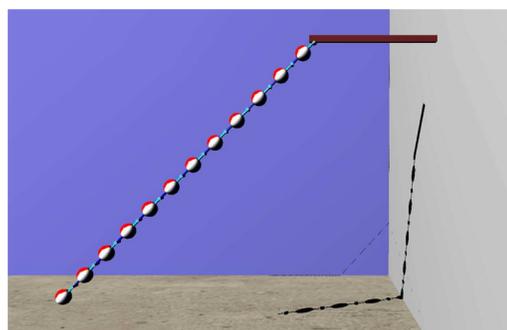


Fig. 13 Initial setup of the 12-link pendulum

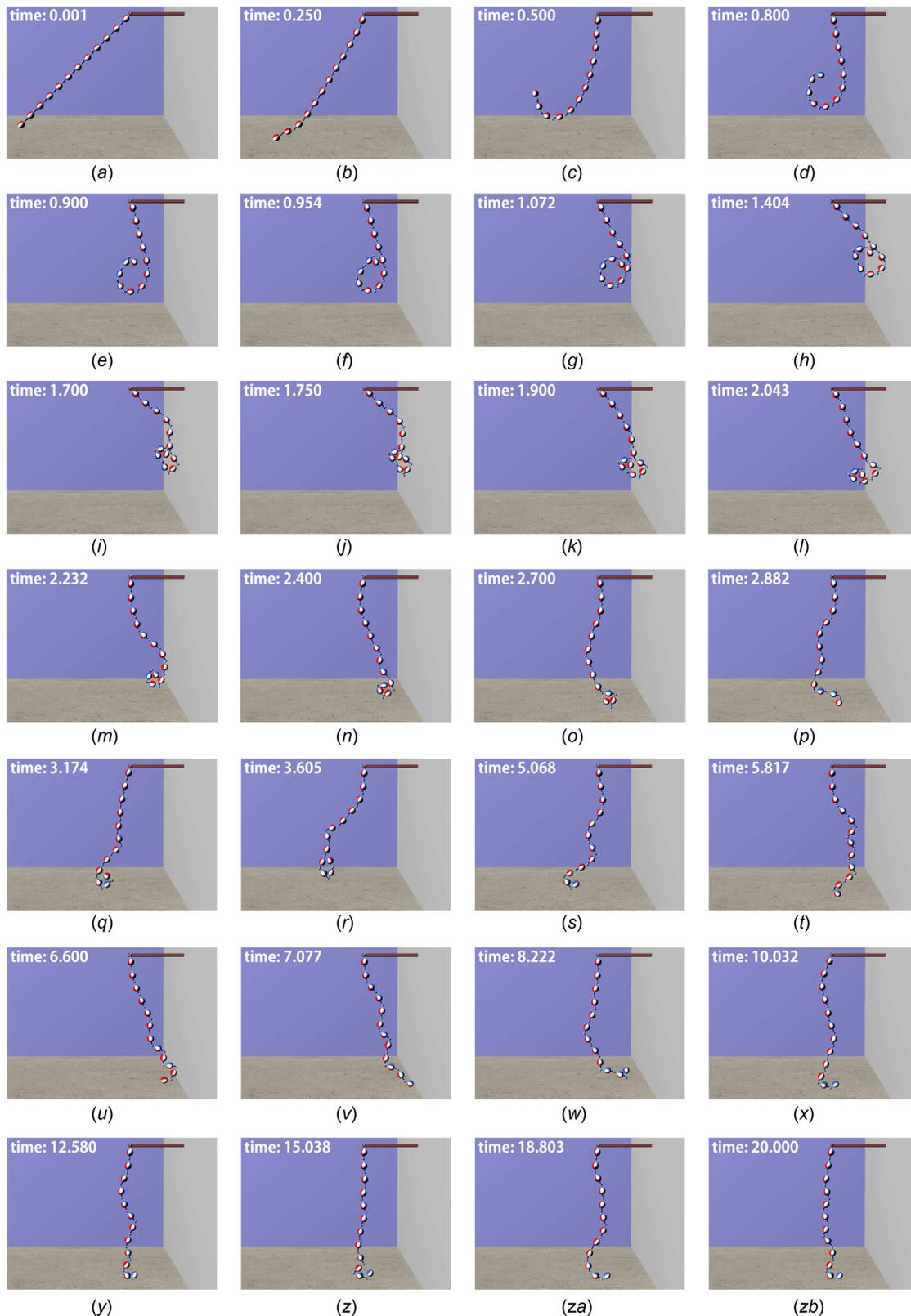


Fig. 14 Snapshots of the motion of a 12-link swinging pendulum from $t = 0$ to $t = 20$ s. Through the course of the simulation, the pendulum collides with itself and its surrounding environment.

Case 1—Friction cone direction vector aligned with tangential friction impulse (validation). Figures 11(a) and 11(b) show time-history plots of the velocities of the sphere and the friction forces in contact coordinates, respectively, computed using the MLCP-PFB-RLM, MLCP-PFB-PLM, MNCP-RLM, and the

MNCP-PLM solvers. We observe that the sphere initially slides in the x -direction until $t_{sr} = [(2v_{x_0})/(7\mu g)] \approx 0.291$ s while gradually losing translational velocity in the x -direction and gaining angular velocity in the y -direction during this time (see Fig. 11(a)). After $t_{sr} = 0.291$ s, the sphere starts rolling with a

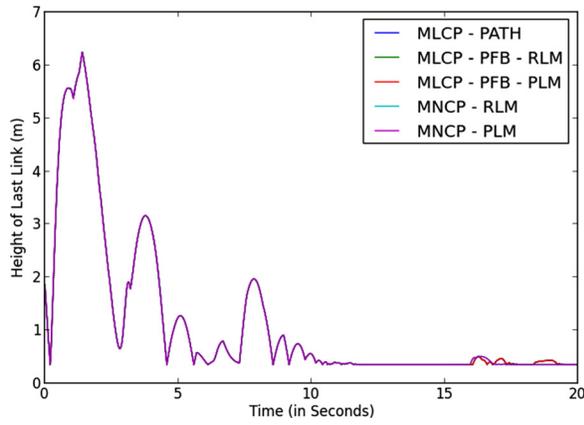


Fig. 15 A time-history plot of the height of the last link of the pendulum computed using the five different solvers. The plots of the five solvers are superposed on top of one another with minor differences showing up at the end of the 20 s simulation. The height of the last link is measured from the origin, which is located at the midsection of the floor (of thickness 0.2 m).

constant linear velocity in the x -direction (v_x) and a constant angular velocity in the y -direction of $\omega_y = ((5v_x)/7) \approx 1.429$ units. The remaining velocity components are all zero (see Fig. 11(a)). Sliding and rolling phases can be confirmed by the fact that in Fig. 11(b), we observe that $\mu F_n = \|F_f, F_o\|$ until $t_{sr} = 0.291$ s, after which $\mu F_n > \|F_f, F_o\|$ when the sphere starts rolling. Further, from Fig. 11(b), we observe that the friction force acts in the $-\hat{f}$ (or $-x$) direction along with $F_o \equiv 0$ throughout the simulation. Thus, as can be inferred from the figures, the solutions of both the MLCP as well as the MNCP set of solvers match the predicted analytical solution (described in the Ref. [1]), which serves in validating all five solvers.

Case 2—Friction cone direction vector misaligned with tangential friction impulse. Ideally, the choice of the friction direction vectors in the contact tangent plane should have no bearing on our final solution. However, they do so in the MLCP case. The degree to which the friction direction vectors in the contact tangent plane are aligned with the tangential friction impulse affects the accuracy of the MLCP solution. This can be seen in the present case where the four direction vectors in Case 1 that are all at right angles to each other are rotated by an angle of 45 deg such that none of these vectors are aligned with the tangential friction impulse or the

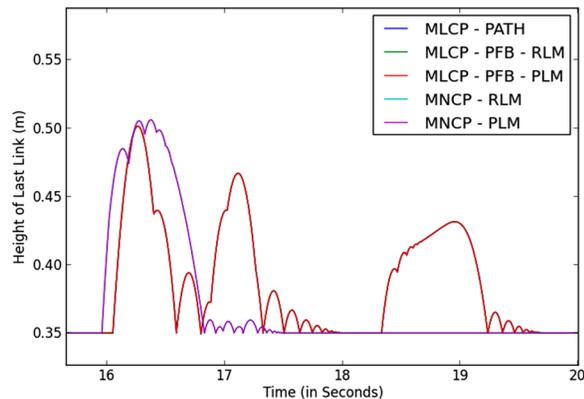


Fig. 17 A zoomed-in plot depicting the height of the last link of the pendulum at the end of the 20s simulation. The solvers show a grouping behavior, with the MLCP set of solvers showing a similar solution and the MNCP set of solvers showing a slightly different solution. The height of the last link is measured from the origin, which is located at the midsection of the floor (of thickness 0.2 m).

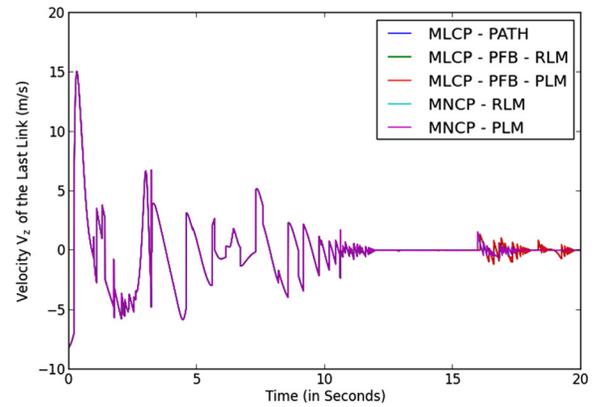


Fig. 16 A time-history plot of the z -component of the linear velocity (v_z) of the last link of the pendulum computed using the five different solvers. The plots of the five solvers are superposed on top of one another with minor differences showing up at the end of the 20 s simulation.

tangential contact velocity (see Fig. 10(b)). The rest of the set up of the problem is exactly the same as the previous case.

From Fig. 12, the MLCP solvers predict that in addition to v_x and ω_y as seen in Case 1, the sphere also possesses nonzero translational velocity v_y and angular velocity ω_x , when in reality, both of these quantities should be zero for all time. Furthermore, the sphere continues to slide until $t \approx 0.4$ s (much longer than the previous case), after which it starts to roll. These results depart considerably from the analytically predicted motion of the sphere seen in Case 1. Thus, any misalignment between the friction direction vectors and the tangential friction impulse vector produces erroneous results in the MLCP case, which is one of the major drawbacks of the approach.

The results from cases 1 and 2 illustrate the importance of aligning (one of) the direction vectors with the opposite of the tangential contact velocity direction (for the MLCP solvers to reduce numerical errors arising from the friction cone discretization). When using MLCP solvers, we thus recommend a two-pronged strategy of (a) aligning one of the direction vectors at the current time step with the opposite of the tangential contact velocity at the same time step and (b) including the opposite direction for each direction vector in the set, to mitigate errors. Further reduction in these errors can be achieved by increasing the number of direction vectors. Increasing the number of direction vectors leads to a more accurate representation of the circular friction limit set at the cost of increasing the size of (and the cost of solving) the MLCP problem.

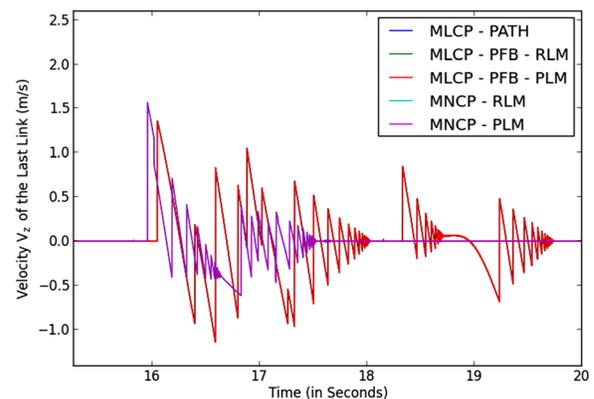


Fig. 18 A zoomed-in plot depicting v_z of the last link of the pendulum at the end of the 20s simulation. The solvers show a grouping behavior, with the MLCP set of solvers showing a similar solution and MNCP set of solvers showing a slightly different solution.

Table 1 Computational times of the five solvers for the 12-link pendulum problem

Complementarity solver	Computational time (s)
MLCP-PATH	23.92
MLCP-PFB-RLM	22.78
MLCP-PFB-PLM	24.24
MNCP-RLM	22.85
MNCP-PLM	22.82

6.2 Twelve-Link Pendulum. To compare the accuracy of the various linear and nonlinear complementarity schemes developed in this study for the case of multiple simultaneous contacts, we consider the example of a 12-link complex pendulum falling under gravity and colliding with its surrounding environment. The pendulum consists of 12 identical 1 kg spherical masses connected together with pin hinges (see Fig. 13). The environment consists of a floor and a wall on the right located 4 m away from the tip of the pendulum. The overall length of the pendulum is 12 m with each of the spheres having a diameter of 0.5 m. The pendulum is located at a height of 10 m above the ground. The coefficient of friction is assumed to be $\mu = 0.5$ and the coefficient of restitution is assumed to be 0.7 for inelastic collisions. The open source software BULLET [35] is used for collision detection. The pendulum makes an initial angle of $\pi/4$ rad with the vertical and has an initial angular velocity of $\omega_x = 1$ rad/s. Uniform gravitational acceleration of 9.81 m/s^2 is assumed.

For the MLCP set of solvers, the number of direction vectors in the contact tangent space is chosen to be four (i.e., $n_f = 4$ in Eq. (30)), with the direction vectors oriented at right angles to each other such that the opposite direction vectors are automatically a part of the set. Furthermore, at each time step, one of the direction vectors at the current time step is aligned with the tangential contact velocity vector at that time step to minimize the friction cone discretization errors (see discussion in Sec. 3.1).

MC OS formulation is used to model the dynamics of this system. Since minimal coordinates are used, the interlink constraints are automatically eliminated. No loop-closure bilateral constraints exist for the multilink pendulum. Hence, the only constraints acting on the pendulum system are unilateral contact constraints. The size of the complementarity problem depends on the number of contacts, with the size of the MNCP problem being smaller compared to the MLCP problem ($3n_u$ for the MNCP versus $n_u(n_f + 2) = 6n_u$ for the MLCP). Except for the PATH-based solver, the ensuing complementarity problems are recast as optimization problems using the methodologies presented in this paper and solved using the RLM and PLM optimization routines.

The simulations are run for a time span of $t = 20$ s with a time step of $\Delta t = 1$ ms. As the pendulum swings from left to right, it collides with the ground, bounces off of the ground, collides with the wall on the right, swings back, and collides with the ground once again. Over the course of the simulation, multiple links are at times in collision with ground, the wall, and with each other. Snapshots of the motion of the pendulum at different time instances are shown in Fig. 14.

Figures 15 and 16 depict the time-history plots of the last link's height and the linear z-velocity (v_z), respectively, computed using the MLCP-PATH, MLCP-PFB-RLM, MLCP-PFB-PLM, MNCP-RLM, and MNCP-PLM solvers. Figures 17 and 18, on the other hand, depict the zoomed-in plots of Figs. 15 and 16, respectively, at the end of the 20 s simulation. As can be seen from figures, the plots show a close match between the solutions of all the five solvers with minor differences appearing at the end of the 20 s simulation. From these simulation results, we deduce that all five complementarity solvers show similar performance in terms of accuracy.

The computation times of the solvers for this 12-link pendulum problem are by and large evenly matched as shown in Table 1. While it is premature to draw a broader conclusion from just this class of examples, we conjecture that this may be more broadly true for the MC formulation. The reason for this is that the size of the complementarity problem is small in the MC approach, and the overall solution cost is dominated by the cost of setting up the complementarity problem rather than the cost of solving the complementarity problem. Furthermore, for our MLCP set of solvers, our two-pronged strategy of including the opposite direction vectors in the set and aligning one of the direction vectors at the current time step with the opposite of the tangential contact velocity vector at the same time step appears to be working well as the results generated by the MLCP set of solvers appear to closely match the results generated by the MNCP set of solvers.

7 Conclusions

In this paper, contact and collision dynamics of articulated rigid multibody systems is approached using the MC OS formulation. The dynamics is cast as a complementarity problem, which is further reformulated as an unconstrained optimization problem. LM-type algorithms are employed to solve these optimization problems. Using these techniques, three linear (MLCP-PFB-RLM, MLCP-PFB-PLM, and MLCP-PATH) and two nonlinear (MNCP-RLM and MNCP-PLM) complementarity schemes have been developed for solving general contact dynamics problems. These solvers have been validated using the example of a sphere moving on a fixed horizontal plane, for which the analytical solutions are available [1].

Furthermore, a 12-link complex pendulum example is used to evaluate the accuracy of the solvers for the case of multiple simultaneous contacts. We found that the four schemes developed in this paper show similar accuracy and closely match the results of the MLCP-PATH algorithm. We further observed similar computational speed for the different solvers and speculate that this may be more broadly true for the MC approach, given that the size of the complementarity problem is small for the formulation and the overall solution cost is dominated by the cost of setting up the complementarity problem (as opposed to solving the problem). In future work, we plan to investigate this conjecture and determine if the cost savings arising from reducing the size of the complementarity problem outweigh the additional costs of setting up the complementarity problem. We intend to perform this study on a larger class of multilink contact dynamics problems.

We have further observed that accurate solutions can be obtained with linear complementarity solvers by aligning one of the friction cone direction vectors in the contact tangent plane (at the current integration time step) to coincide with the opposite of the tangential contact velocity vector at that time step (see discussion in Sec. 3.1). Using such an alignment procedure, it is possible to use linear, rather than nonlinear complementarity solvers, with just a small number of friction cone direction vectors without incurring large accuracy or cost penalties.

Acknowledgment

The research described in this paper was performed at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with the National Aeronautics and Space Administration.¹

¹©2015 California Institute of Technology. Government sponsorship acknowledged.

Nomenclature

- FB = Fischer–Burmeister function
LCP = linear complementarity problem
LM = Levenberg–Marquardt algorithm
MC = minimal coordinate formulation
MLCP = mixed linear complementarity problem
MLCP–PATH = contact dynamics solver, where the dynamics is formulated as an MLCP and solved using the PATH [7] algorithm
MLCP–PFB–PLM = contact dynamics solver, where the dynamics is formulated as an MLCP, which is further recast as an optimization problem using the PFB function, and solved with the help of the RLM algorithm
MLCP–PFB–RLM = contact dynamics solver, where the dynamics is formulated as an MLCP, which is further recast as an optimization problem using the PFB function, and solved with the help of the RLM algorithm
MNCP = mixed nonlinear complementarity problem
MNCP–PLM = similar to the MNCP–RLM solver except that the PLM algorithm is used to solve the optimization problem
MNCP–RLM = contact dynamics solver, where the dynamics is formulated as an MNCP using Todorov’s approach [8] (the MNCP is recast as an optimization problem and solved using the RLM algorithm)
NCP = nonlinear complementarity problem
OS = operational space formulation
PFB = penalized Fischer–Burmeister function
PLM = projected Levenberg–Marquardt algorithm
RC = redundant coordinate formulation
RLM = regular Levenberg–Marquardt algorithm

References

- [1] Trinkle, J., 2003, “Formulation of Multibody Dynamics as Complementarity Problems,” *ASME Paper No. DETC2003/VIB-48342*.
- [2] Todorov, E., Erez, T., and Tassa, Y., 2012, “Mujoco: A Physics Engine for Model-Based Control,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, Oct. 7–12, pp. 5026–5033.
- [3] Jain, A., 2014, “Contact Dynamics Formulation Using Minimal Coordinates,” *Multibody Dynamics*, Springer International Publishing, Switzerland, pp. 93–121.
- [4] Anitescu, M., and Potra, F. A., 1997, “Formulating Dynamic Multi-Rigid-Body Contact Problems With Friction as Solvable Linear Complementarity Problems,” *Nonlinear Dyn.*, **14**(3), pp. 231–247.
- [5] Murty, K. G., 1988, *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann, Berlin.
- [6] Tasora, A., and Anitescu, M., 2009, “A Fast NCP Solver for Large Rigid-Body Problems With Contacts, Friction, and Joints,” *Multibody Dynamics*, Springer, The Netherlands, pp. 45–55.
- [7] Dirkse, S., Ferris, M., and Munson, T., 2013, “The PATH Solver.”
- [8] Todorov, E., 2010, “Implicit Nonlinear Complementarity: A New Approach to Contact Dynamics,” *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, May 3–7, pp. 2322–2329.
- [9] Kanzow, C., and Petra, S., 2004, “On a Semismooth Least Squares Formulation of Complementarity Problems With Gap Reduction,” *Optim. Methods Software*, **19**(5), pp. 507–525.
- [10] Kanzow, C., and Petra, S., 2005, “LMMCP: A Levenberg–Marquardt Type MATLAB Solver for Mixed Complementarity Problems.”
- [11] Kanzow, C., and Petra, S., 2005, “Projected Filter Trust Region Methods for a Semismooth Least Squares Formulation of Mixed Complementarity Problems,” *Optim. Methods Software*, **22**(5), pp. 713–735.
- [12] Trinkle, J. C., Pang, J.-S., Sudarsky, S., and Lo, G., 1997, “On Dynamic Multi-Rigid-Body Contact Problems With Coulomb Friction,” *Z. Angew. Math. Mech.*, **77**(4), pp. 267–279.
- [13] Jain, A., 2011, *Robot and Multibody Dynamics: Analysis and Algorithms*, Springer, New York.
- [14] Jain, A., Crean, C., Kuo, C., Von Bremen, H., and Myint, S., 2012, “Minimal Coordinate Formulation of Contact Dynamics in Operational Space,” *Robotics: Science and Systems*, MIT Press, Robotics: Science and Systems, Sydney, Australia, July.
- [15] Lacoursiere, C., Lu, Y., Williams, J., and Trinkle, J., 2013, “Standard Interface for Data Analysis of Solvers in Multibody Dynamics,” 4th Canadian Conference on Nonlinear Solid Mechanics (CanCNSM 2013), Montreal, QC, Canada, July 23–26, Vol. 8, Paper No. 814.
- [16] Glocker, C., 2013, “Simulation of Hard Contacts With Friction: An Iterative Projection Method,” *Recent Trends in Dynamical Systems*, Springer, Basel, Switzerland, pp. 493–515.
- [17] Chen, B., Chen, X., and Kanzow, C., 2000, “A Penalized Fischer–Burmeister NCP-Function,” *Math. Program.*, **88**(1), pp. 211–216.
- [18] Negrut, D., Tasora, A., Mazhar, H., Heyn, T., and Hahn, P., 2012, “Leveraging Parallel Computing in Multibody Dynamics,” *Multibody Syst. Dyn.*, **27**(1), pp. 95–117.
- [19] Marquardt, D. W., 1963, “An Algorithm for Least-Squares Estimation of Nonlinear Parameters,” *J. Soc. Ind. Appl. Math.*, **11**(2), pp. 431–441.
- [20] Bazaraa, M. S., Sherali, H., and Shetty, C., 1993, *Nonlinear Programming: Theory and Algorithms*, Wiley, Hoboken, NJ.
- [21] Mylapilli, H., and Jain, A., 2014, “Evaluation of Complementarity Techniques for Minimal Coordinate Contact Dynamics,” *ASME Paper No. DETC2014-34322*.
- [22] Jain, A., 2014, “Operational Space Inertia for Closed-Chain Robotic Systems,” *ASME J. Comput. Nonlinear Dyn.*, **9**(2), p. 021015.
- [23] Geiger, C., and Kanzow, C., 1996, “On the Resolution of Monotone Complementarity Problems,” *Comput. Optim. Appl.*, **5**(2), pp. 155–173.
- [24] Studer, C. W., 2008, “Augmented Time-Stepping Integration of Non-Smooth Dynamical Systems,” *Ph.D. thesis*, ETH, Zürich, Switzerland.
- [25] Sun, D., and Qi, L., 1999, “On NCP-Functions,” *Comput. Optim. Appl.*, **13**(1–3), pp. 201–220.
- [26] Fischer, A., 1995, “A Newton-Type Method for Positive-Semidefinite Linear Complementarity Problems,” *J. Optim. Theory Appl.*, **86**(3), pp. 585–608.
- [27] Ferris, M. C., Kanzow, C., and Munson, T. S., 1999, “Feasible Descent Algorithms for Mixed Complementarity Problems,” *Math. Program.*, **86**(3), pp. 475–497.
- [28] Jain, A., Crean, C., Kuo, C., and Quadrelli, M. B., 2012, “Efficient Constraint Modeling for Closed-Chain Dynamics,” *2nd Joint International Conference on Multibody System Dynamics*, Stuttgart, Germany, May 29–June 1.
- [29] Schindler, T., and Acary, V., 2014, “Timestepping Schemes for Nonsmooth Dynamics Based on Discontinuous Galerkin Methods: Definition and Outlook,” *Math. Comput. Simul.*, **95**, pp. 180–199.
- [30] Schoeder, S., Ulbrich, H., and Schindler, T., 2014, “Discussion of the Gear–Gupta–Leimkuhler Method for Impacting Mechanical Systems,” *Multibody Syst. Dyn.*, **31**(4), pp. 477–495.
- [31] Drumwright, E., and Shell, D. A., 2009, “A Robust and Tractable Contact Model for Dynamic Robotic Simulation,” *ACM Symposium on Applied Computing*, Honolulu, HI, Mar. 8–12, pp. 1176–1180.
- [32] Boyd, S. P., and Vandenberghe, L., 2004, *Convex Optimization*, Cambridge University Press, New York.
- [33] Grippo, L., Lampariello, F., and Lucidi, S., 1986, “A Nonmonotone Line Search Technique for Newton’s Method,” *SIAM J. Numer. Anal.*, **23**(4), pp. 707–716.
- [34] Grippo, L., Lampariello, F., and Lucidi, S., 1991, “A Class of Nonmonotone Stabilization Methods in Unconstrained Optimization,” *Numer. Math.*, **59**(1), pp. 779–805.
- [35] Coumans, E., 2013, “Bullet Physics Library.”