ECCOMAS Multibody Dynamics 2013
1-4 July, 2013, University of Zagreb, Croatia

Minimal Coordinates Formulation of Contact Dynamics

Abhinandan Jain[1]

[1] Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA, 91109 USA, Abhi.Jain@jpl.nasa.gov

Abstract

In recent years, complementarity techniques have been developed for modeling non-smooth dynamics arising from contact and collision problems for multi-link robotic systems. The commonly used complementarity approach sets up a linear complementarity problem (LCP) using non-minimal coordinates together with all the unilateral contact constraints and inter-link hinge and loop closure bilateral constraints on the system. In this paper, we develop a complementarity formulation that uses an operational space approach. It uses minimal coordinates resulting in a much smaller LCP problem whose size is independent of the number of bodies and the number of degrees of freedom in the system. Furthermore, we exploit operational space low-order algorithms to overcome key computational bottlenecks to obtain over an order of magnitude speed up in the solution procedure.

Keywords: *robotics, multibody dynamics, non-smooth dynamics*

1 Introduction

For more than a decade, researchers have been developing complementarity based approaches for formulating and solving the equations of motion of systems with contact and collision dynamics [1–3]. Examples of such dynamics for robotic systems include manipulation and grasping tasks such as illustrated in Figure 1, and legged locomotion. The complementarity approach models bodies as rigid, and uses impulsive dynamics to handle non-smooth collision, contact interactions and mode transitions. By impulsively “stepping” over non-smooth events, complementarity methods avoid small time step size and stiffening issues encountered with penalty based methods which allow surface compliance during contact [4]. In this paper, we focus on the analytical and computational aspects of a minimal coordinate formulation of the complementarity approach to contact and collision dynamics for multi-link systems. This paper further extends the operational space formulation for contact and collision dynamics described in reference [5]. We adopt the complementarity based physics models from [2, 3].

The complementarity based solution consists of a combination of: (a) setting up a *linear complementarity problem (LCP)* problem; (b) numerically solving the LCP problem; and (c) ancillary dynamics computations. The LCP depends on the link mass and inertia properties, contact friction parameters, inter-link bilateral constraints and contact and collision unilateral constraints. The LCP solution identifies the unilateral constraints that are active, and solves for the impulsive forces and velocity changes that are consistent with the constraints on the system. Variants of the complementarity approach to handle elastic and inelastic collisions have also been developed [3]. While LCP formulations use discretized approximations for the friction cones, other researchers have explored non-linear cone complementarity approaches that avoid such approximations [6].

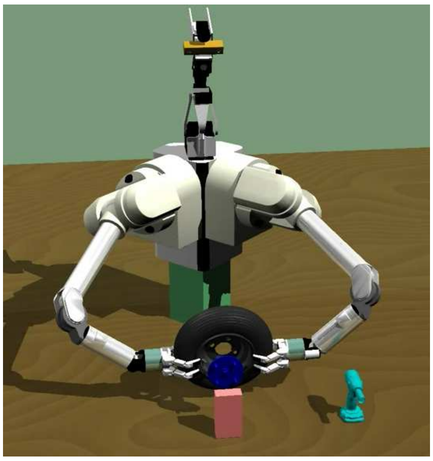

Figure 1. An example multi-arm robot manipulation task involving unmounting a wheel from a hub involving several contact and collision dynamics interaction events.

The typical approach to handling contact and collision dynamics is to work with non-minimal coordinates, since it is the easiest to set up [3]. In this formulation, the LCP stage does most of the work, but the LCP dimension is large and computationally expensive to solve. For a multi-link system with n links, the standard LCP approach uses $6n$ non-minimal coordinates together with the bilateral constraints associated with the inter-link hinges and the unilateral contact constraints. This approach has a mass matrix that is block diagonal and constant. Besides the large LCP problem size, this formulation requires additional techniques for managing error drift in the bilateral constraints when integrating the equations of motion.

An alternative approach is to use minimal hinge coordinates that automatically eliminate the bilateral constraints for the inter-link hinges [7]. While the underlying physics remains unchanged, this formulation reduces the size of the LCP problem, and avoids the need for managing bilateral constraint violation errors for the hinges. However, the use of minimal coordinates leads to a dense and configuration dependent mass matrix. Thus while minimal coordinates lead to smaller LCP problems, they also typically significantly increase the difficulty and computational complexity of setting up the LCP problem. This has been a significant hurdle in the use of minimal coordinate approaches.

In this paper we explore a progression of minimal coordinate formulations that partition the overall solution effort in different ways between setting up the LCP problem, and solving it. Our goal is to reduce the overall computational complexity by taking advantage of the smaller dimension of minimal coordinate models together with the host of structure based, recursive and low-order dynamics algorithms that are available for articulated system dynamics. Notable examples of such structure based algorithms include the composite rigid body algorithms for computing the mass matrix [8], the articulated body inertia forward dynamics algorithm [9] and the spatial operator based operational space dynamics algorithm [10].

The main contribution of this paper is in the development of an operational space based *OS formulation*, that while using minimal coordinates for the contact and collision dynamics problem, uses low-order spatial operator algorithms to overcome the complexitys of setting up the LCP problem. This results in a more than an order of magnitude reduction in computational complexity. The size of the resulting LCP problem is independent of the number of links and generalized coordinates, and only depends on the number of contact nodes. We also describe extensions of the formulation to handle elastic and inelastic collision dynamics. The different formulations are developed and described in a way so as to clarify the relationships among them and methods available in the literature.

We use a multi-link pendulum numerical problem to quantitatively and parameterically measure the performance improvements from the new OS formulation. We have also applied the OS formulation to simulate manipulations tasks involving contact and collision interactions for a dual-arm robot system. This dual-arm robot is used as a reference system to compare the LCP sizes for the different formulations discussed in this article.

The organization of this paper is as follows. Section 2 describes the complementarity conditions associated with modeling a single unilateral contact constraint. In Section 3 we describe the system-level, multiple contacts *NMC* LCP formulation based on non-minimal coordinates that is widely used. While easy to set up, this formulation leads to a large LCP problem. Section 4 discusses a similar *MC* formulation based on the use of minimal coordinates. We observe that the reduction in the size of the LCP is accompanied by an increase in the complexity of setting up the LCP. Section 5 further transforms the MC LCP formulation to develop the *RMC* formulation that further reduces the size of the LCP problem, but at the cost of an increase in the LCP setup complexity. Section 6 develops the *OS* formulation that is based on an operational space approach. While the LCP size is moderately larger than from the RMC approach, it opens the path to exploiting low-order operational space algorithms to significantly reduce the LCP setup complexity. Section 7 extends the OS formulation contact dynamics model to include elastic and inelastic collision dynamics. Section 8 focuses on computational issues, and describes operational space computational algorithms to reduce the complexity of setting up the OS LCP problem. The section also includes results from a numerical simulation to quantify the performance improvements for the OS formulation.

2 Unilateral contact constraints

Unilateral constraints are defined by inequality relationships of the form

$$\eth(\theta, t) \geqslant 0 \tag{1}$$

for some function $\eth$ of the configuration coordinates θ and time t. As an example, the non-penetration condition for rigid bodies can be stated as an inequality relationship requiring that the distance between the surfaces of rigid bodies be non-negative. $\eth(\theta, t)$ is generally referred to as the *distance* or *gap* function.

Contact occurs at the constraint boundary, i.e., when $\eth(\theta, t) = 0$. For bodies in contact, the surface normals at the contact point are parallel. The existence of contact is typically determined using geometric or collision detection techniques. For a pair of bodies A and B in contact, we use a convention where the i^{th} contact normal $\hat{n}(i)$ is defined as pointing from body B towards body A, so that motion of A in the direction of the normal leads to a separation of the bodies. A unilateral constraint is said to be in an *active* state when

$$\eth(\theta, t) = \dot{\eth}(\theta, t) = \ddot{\eth}(\theta, t) = 0 \tag{2}$$

Thus, a unilateral constraint is active when there is contact, and the contact persists. Only active constraints generate constraint forces on the system. A constraint that is not active is said to be *inactive*. Contact *separation* occurs when the relative linear velocity of the contact points along the normal 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

$$\eth(\theta, t) = \dot{\eth}(\theta, t) = 0 \quad \text{and} \quad \ddot{\eth}(\theta, t) > 0 \tag{3}$$

2.1 Contact impulse for an active contact constraint

We now describe contact force modeling using the approach in references [2, 3]. The 6-dimensional spatial impulse at the i^{th} active contact constraint node has a zero angular moment component. Its non-zero linear impulse component $F_u(i) \in \mathcal{R}^3$ can be decomposed into normal and tangential (friction impulse) components

$$F_u(i) = F_n(i)\hat{n}(i) + F_t(i)\hat{t}(i) \tag{4}$$

where $\hat{t}(i)$ denotes a tangent plane vector for the i^{th} contact pair. Assuming that the friction coefficient is $\mu(i)$, the magnitude of the tangential Coulomb frictional impulse is bounded by the magnitude of the normal component as follows:

$$\|F_t(i)\| \leqslant \mu(i)F_n(i) \tag{5}$$

When the bodies have non-zero relative linear velocities at the contact point, the contact is said to be a *sliding* contact. Otherwise, when the relative linear velocity is zero, the contact is said to be a *rolling* contact. During sliding, the tangential frictional impulse is in a direction opposing the linear velocity vector (which necessarily lies in the contact tangent plane) and Eq. 5 holds with an equality. Thus, the tangential friction impulse is on the boundary of the cone defined by Eq. 5 when sliding, and in the interior of the cone when rolling.

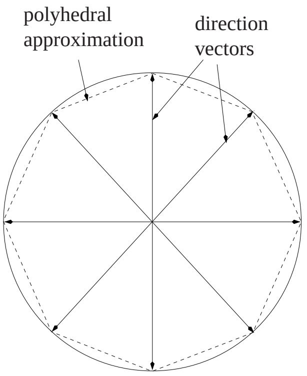

Figure 2. Polyhedral approximation of the friction cone.

For the purpose of numerical computation, 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 (see Figure 2.1). It is assumed that for each direction vector, its opposite direction vector is also in the set. For notational simplicity, we assume that n_f is the same across all contact points. The i^{th} contact tangential frictional impulse is expressed as the linear combination of these direction vectors as follows:

$$F_t(i)\hat{t}(i) = \sum_{j=1}^{n_f} \beta_j(i)\hat{d}_j(i) = D(i)\beta(i) \tag{6}$$

where

$$D(i) \triangleq \left[\hat{d}_1(i), \cdots, \hat{d}_{n_f}(i)\right] \in \mathcal{R}^{3\times n_f} \quad \text{and} \quad \beta(i) \triangleq \text{col}\{\beta_j(i)\}_{j=1}^{n_f} \in \mathcal{R}^{n_f}$$

Combining Eq. 4 and Eq. 6 we have

$$F_u(i) = \underline{D}(i)\underline{\beta}(i), \quad \text{where} \quad \underline{\beta}(i) \triangleq \begin{bmatrix} F_n(i) \\ \beta(i) \end{bmatrix} \in \mathcal{R}^{n_f+1} \quad \text{and} \quad \underline{D}(i) \triangleq \left[\hat{n}(i),\ D(i)\right] \in \mathcal{R}^{3\times(n_f+1)} \tag{7}$$

During sliding, the $\beta_j(i)$ component is non-zero and 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. In other words, with $\sigma(i)$ denoting the magnitude of the contact relative linear velocity,

$$\beta_k(i) = \begin{cases} \mu(i)F_n(i)\mathbb{1}_{[k=j]} & \text{if} \quad \sigma(i) > 0 \\ 0 & \text{if} \quad \sigma(i) = 0 \end{cases} \tag{8}$$

In the above, $\mathbb{1}_{[<cond>]}$ denotes the indicator function whose value is 1 if the condition is true, and 0 otherwise.

2.2 Complementarity relationship for a unilateral contact

We begin by defining complementarity conditions. Let $f(z) \in \mathcal{R}^n$ denote a function of a vector $z \in \mathcal{R}^n$, whose z_i elements have lower and upper bounds l_i and u_i respectively. The *complementarity condition*, $f(z) \perp z$, is said to hold when the following properties apply:

- $f_i(z) \geqslant 0$ when $z_i = l_i$
- $f_i(z) \leqslant 0$ when $z_i = u_i$
- $f_i(z) = 0$ when $l_i < z_i < u_i$

Typically the bounds are $l_i = 0$ and $u_i = \infty$, and we will assume this to be the case unless otherwise stated. For these bounds, the elements of $f(z)$ and z are non-negative, and the complementarity condition requires that for any i, only one of f_i or z_i can be positive. A complementarity condition is a *linear complementarity condition* when $f(z)$ has the form $\mathfrak{M}z + q \quad \perp \quad z$ for some matrix $\mathfrak{M}$ and vector q. Thus for an LCP

$$\mathfrak{M}\, z + q \quad \perp \quad z \tag{9}$$

We have a *mixed complementarity condition* when one or more of the rows of $f(z)$ are exactly equal to zero, i.e. the bounds for one or more of the rows are $l_i = -\infty$ and $u_i = \infty$. Such identically zero rows represent equality conditions while the remainder are complementarity (inequality) conditions.

The sliding/rolling contact relationships described above can be rephrased as the following complementarity conditions[1]:

$$\begin{aligned}
\hat{n}^*(i)v_u^+(i) \quad &\perp \quad F_n(i) \quad \textit{(separation)} && (10a)\\
\sigma(i)E(i) + D^*(i)v_u^+(i) \quad &\perp \quad \beta(i) \quad \textit{(friction force direction)} && (10b)\\
\mu(i)F_n(i) - E^*(i)\beta(i) \quad &\perp \quad \sigma(i) \quad \textit{(friction force magnitude)} && (10c)
\end{aligned}$$

where

$$E(i) \;\stackrel{\triangle}{=}\; \text{col}\{1\}_{j=1}^{n_f} \in \mathcal{R}^{n_f} \tag{11}$$

and $v_u^+(i) \in \mathcal{R}^3$ denotes the relative linear velocity of the first body A with respect to the second body B. The component of this relative linear velocity along the contact normal is, $\hat{n}^*(i)v_u^+(i)$. A positive value implies increasing separation between the bodies, while a negative value indicates that the bodies are approaching each other. Eq. 10a states that this velocity component and the normal interaction impulse $F_n(i)$ cannot both be simultaneously positive. Thus the interaction impulse must be zero when the bodies are separating, and the impulse can be non-zero only if we have sustained contact. Eq. 10b implies that the tangential friction impulse opposes the tangential relative linear velocity, while Eq. 10c states that the magnitude of the tangential impulse is on the friction cone boundary when the the tangential relative linear velocity is non-zero.

The complementarity conditions in Eq. 10 enforce the no inter-penetration constraint at the velocity instead of at the gap level. Hence they are valid only when the gap is zero, i.e., when contact exists [3]. Using Eq. 7, Eq. 10 can be expressed more compactly as

$$\begin{aligned}
\hat{E}(i)\sigma(i) + \underline{D}^*(i)v_u^+(i) \quad &\perp \quad \underline{\beta}(i)\\
\overline{E}(i)\underline{\beta}(i) \quad &\perp \quad \sigma(i)
\end{aligned} \tag{12}$$

where

$$\hat{E}(i) \;\stackrel{\triangle}{=}\; \begin{bmatrix} 0 \\ E(i) \end{bmatrix} \in \mathcal{R}^{(n_f+1)} \quad \text{and} \quad \overline{E}(i) \;\stackrel{\triangle}{=}\; [\mu(i),\ -E^*(i)] \in \mathcal{R}^{1\times(n_f+1)} \tag{13}$$

With n_u denoting the number of unilateral contact nodes, the component level complementarity conditions in Eq. 12 can be aggregated across all the contact constraints and expressed at the system level as:

$$\hat{E}\sigma + \underline{D}^* v_u^+ \quad \perp \quad \underline{\beta} \qquad \text{and} \qquad \overline{E}\underline{\beta} \quad \perp \quad \sigma \tag{14}$$

where

$$\begin{aligned}
\underline{\beta} \;&\stackrel{\triangle}{=}\; \text{col}\left\{\underline{\beta}(i)\right\}_{i=1}^{n_u} \in \mathcal{R}^{n_u(n_f+1)}, & \sigma \;&\stackrel{\triangle}{=}\; \text{col}\{\sigma(i)\}_{i=1}^{n_u} \in \mathcal{R}^{n_u}\\
\underline{D} \;&\stackrel{\triangle}{=}\; \text{diag}\{D(i)\}_{i=1}^{n_u} \in \mathcal{R}^{3n_u\times n_u(n_f+1)}, & \hat{E} \;&\stackrel{\triangle}{=}\; \text{diag}\left\{\hat{E}(i)\right\}_{i=1}^{n_u} \in \mathcal{R}^{n_u(n_f+1)\times n_u}\\
\text{and} \quad \overline{E} \;&\stackrel{\triangle}{=}\; \text{diag}\left\{\overline{E}(i)\right\}_{i=1}^{n_u} \in \mathcal{R}^{n_u\times n_u(n_f+1)}, & v_u^+ \;&\stackrel{\triangle}{=}\; \text{col}\left\{v_u^+(i)\right\}_{i=1}^{n_u} \in \mathcal{R}^{3n_u}
\end{aligned} \tag{15}$$

Also, Eq. 7 can be restated at the system level as

$$F_u = \underline{D}\underline{\beta} \quad \text{where} \quad F_u \;\stackrel{\triangle}{=}\; \text{col}\{F_u(i)\}_{i=1}^{n_u} \in \mathcal{R}^{3n_u} \tag{16}$$

[1] For a vector/matrix A, the A^* notation denotes its vector/matrix transpose.

3 Non-minimal coordinates (NMC) LCP formulation

In this section we derive the commonly used non-minimal coordinate LCP formulation for contact dynamics based on the approach in [3]. We refer to this formulation as the *non-minimal coordinates (NMC)* formulation.

Contact and collision dynamics models build upon smooth dynamics models. The smooth dynamics model used by the NMC method treats all the links in the system as independent bodies, and all coupling hinges as explicit bilateral constraints as illustrated in Figure 3. Such a smooth dynamics model utilizes non-minimal coordinates and is also referred to as a *fully augmented (FA)* model [11].

Let $\mathfrak{n}$ denote the number of links in the system, and $\mathcal{N}$ the number of system degrees of freedom in the absence of bilateral constraints. For the FA model $\mathcal{N} = 6\mathfrak{n}$. Let $\mathfrak{n}_b$ denote the dimension of the bilateral constraints arising from inter-link hinges and loop closure constraints on the system. With x denoting the vector of positional and attitude coordinates for the links, let $\mathcal{V} \in \mathcal{R}^{6\mathfrak{n}}$ denote the stacked vector of spatial velocities of all the links. Then there exists a $G_b(x,t) \in \mathcal{R}^{\mathfrak{n}_b \times 6\mathfrak{n}}$ matrix and a $\mathfrak{U}(t) \in \mathcal{R}^{\mathfrak{n}_b}$ vector that define the following velocity domain constraint equation for the bilateral constraints on the system:

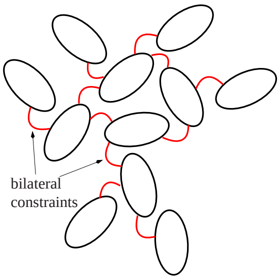

Figure 3. Fully augmented model with hinges modeled as constraints.

$$G_b(x,t)\mathcal{V} = \mathfrak{U}(t) \tag{17}$$

We assume that $G_b(x,t)$ is a *full-rank* matrix. Observe that Eq. 17 is linear in $\mathcal{V}$. The bilateral constraints effectively reduce the independent degrees of freedom for the system from $\mathcal{N}$ to $(\mathcal{N} - \mathfrak{n}_b)$. The bilateral constraints are accounted for via *Lagrange multipliers*, $\lambda \in \mathcal{R}^{\mathfrak{n}_b}$ to yield the following smooth equations of motion for the system

$$\begin{gathered} M\alpha - G_b^*(x,t)\lambda = C(x,\mathcal{V}) \\ G_b(x,t)\mathcal{V} = \mathfrak{U}(t) \end{gathered} \tag{18}$$

where $\alpha \in \mathcal{R}^{6\mathfrak{n}}$ denotes the spatial acceleration of the bodies. $M \in \mathcal{R}^{6\mathfrak{n}\times 6\mathfrak{n}}$ is a block diagonal matrix with the 6×6 spatial inertias of each of the links along the diagonal. $C \in \mathcal{R}^{6\mathfrak{n}}$ is a vector of the velocity dependent Coriolis and external forces on the system. The $-G_b^*(x,t)\lambda$ term in the first equation represents the constraint forces from the bilateral constraints. Differentiating the Eq. 17 constraint equation, Eq. 18 can be rearranged into the following descriptor form:

$$\begin{pmatrix} M & -G_b^* \\ G_b & 0 \end{pmatrix} \begin{bmatrix} \alpha \\ \lambda \end{bmatrix} = \begin{bmatrix} C \\ \overline{\mathfrak{U}} \end{bmatrix} \quad \text{where} \quad \overline{\mathfrak{U}} \overset{\triangle}{=} \dot{\mathfrak{U}} - \dot{G}_b\mathcal{V} \in \mathcal{R}^{\mathfrak{n}_b} \tag{19}$$

An attractive feature of these smooth equations of motion is that the M matrix is block diagonal and constant. Using the following discrete time Euler step approximation over a Δ_t time interval,[2]

$$\mathcal{V}^+ - \mathcal{V}^- = \alpha\Delta_t \quad \text{and} \quad p_b \overset{\triangle}{=} \lambda\Delta_t \in \mathcal{R}^{\mathfrak{n}_b} \tag{20}$$

the differential form of the equations of motion in Eq. 19 can be transformed into the following discretized version that maps the p_b impulse stacked vector at the bilateral constraint nodes into the resulting change in body spatial velocities.

$$\begin{pmatrix} M & -G_b^* \\ G_b & 0 \end{pmatrix} \begin{bmatrix} \mathcal{V}^+ - \mathcal{V}^- \\ p_b \end{bmatrix} = \begin{bmatrix} C\Delta_t \\ \overline{\mathfrak{U}}\Delta_t \end{bmatrix} \tag{21}$$

3.1 Including contact impulses

The stacked vector of relative linear velocities across the contact nodes is denoted $\nu_u \in \mathcal{R}^{3\mathfrak{n}_u}$. It is related to the stacked vector of body spatial velocities $\mathcal{V}$ via the following relationship

$$\nu_u = G_u\mathcal{V} \tag{22}$$

where the $G_u \in \mathcal{R}^{3\mathfrak{n}_u \times 6\mathfrak{n}}$ matrix contains one block-row per contact node-pair, with each row mapping the spatial velocities for a node pair into the relative linear velocity across the contact. The G_u matrix also relates the F_u equal

[2] The $-$ and $+$ superscripts denote the respective value of a quantity just before and after the application of an impulse.

and opposite impulses at the contact node-pairs to the corresponding spatial impulses on the bodies, $p_u \in \mathcal{R}^{6n}$ via the following dual mapping

$$p_u = G_u^* F_u \tag{23}$$

The p_u contact impulses can be included in the Eq. 21 smooth equations of motion by adding p_u to the $C\Delta_t$ term to obtain

$$\begin{pmatrix} M & -G_b^* \\ G_b & 0 \end{pmatrix} \begin{bmatrix} \mathcal{V}^+ - \mathcal{V}^- \\ p_b \end{bmatrix} = \begin{bmatrix} C\Delta_t + p_u \\ \overline{\mathfrak{U}}\Delta_t \end{bmatrix} \tag{24}$$

3.2 Assembling the system LCP

We now set up an LCP to help solve the equations of motion and the the unknown constraint forces. From Eq. 16 and Eq. 23 we have

$$F_u = \underline{D}\underline{\beta} \quad \Rightarrow \quad p_u = G_u^* \underline{D}\underline{\beta} \tag{25}$$

Thus Eq. 24 can be recast as

$$\begin{pmatrix} M & -G_b^* & -G_u^*\underline{D} \\ G_b & 0 & 0 \end{pmatrix} \begin{bmatrix} \mathcal{V}^+ - \mathcal{V}^- \\ p_b \\ \underline{\beta} \end{bmatrix} = \begin{bmatrix} C\Delta_t \\ \overline{\mathfrak{U}}\Delta_t \end{bmatrix} \tag{26}$$

Combining this with the complementarity conditions in Eq. 14 leads to the following NMC formulation of the LCP in Eq. 9:

$$\mathfrak{M} \triangleq \left(\begin{array}{cc|cc} M & -G_b^* & -G_u^*\underline{D} & 0 \\ G_b & 0 & 0 & 0 \\ \hline \underline{D}^* G_u & 0 & 0 & \hat{E} \\ 0 & 0 & \bar{E} & 0 \end{array}\right), \quad z \triangleq \left[\begin{array}{c} \mathcal{V}^+ \\ p_b \\ \hline \underline{\beta} \\ \sigma \end{array}\right], \quad q \triangleq \left[\begin{array}{c} -M\mathcal{V}^- - C\Delta_t \\ -G_b\mathcal{V}^- - \overline{\mathfrak{U}}\Delta_t \\ \hline 0 \\ 0 \end{array}\right] \tag{27}$$

This is a mixed LCP problem, where the first two rows are equality conditions, while the lower two rows are complementarity conditions. This NMC LCP formulation is essentially the one described in [3]. It makes use of non-minimal coordinates for the articulated system and is of size $(6n + n_b + n_u(n_f + 2))$. The constant and block-diagonal structure of M results in $\mathfrak{M}$ having a simple and highly sparse structure. The complexity of assembling $\mathfrak{M}$ and q for the LCP is just $O(n)$. Reference [3] derives sufficient conditions for the existence of a solution for the LCP problem.

The solution of the Eq. 27 LCP provides new $\mathcal{V}^+$ velocity coordinates which can be numerically integrated to propagate the x configuration coordinates. The solution values of $\underline{\beta}$ indicate which contacts are active or inactive, while the values of σ define the rolling or sliding state of each of the active contacts. Thus an LCP solution with $F_u(i)$ positive indicates that the i^{th} contact is *active*. Furthermore, $\sigma(i) = 0$ implies that the i^{th} contact is a *rolling contact* while a positive value implies that it is a *sliding* contact.

In the NMC formulation, the LCP does virtually all the work, and the complexity of setting up the LCP is relative low. The main disadvantage of this formulation is the large size of the LCP and the consequent large complexity of solving it. Moreover, the use of non-minimal coordinates requires the additional use of constraint error stabilization schemes to avoid the build up of constraint violation errors for the bilateral constraints.

We will use the dual-arm robot in Figure 1 to track and compare the LCP size for this formulation and the ones to follow. This dual-arm platform has a 4 link sensor head, a pair of 7 link arms, with each arm having a 3 finger hand for an overall system with 26 links and 26 degrees of freedom. It has no loop closure bilateral constraints. Thus $n = 26$, $\mathcal{N} = 6n = 156$, and $n_b = 5n = 130$. For this exercise we assume that $n_f = 4$, and that there are 4 contact constraints. With these parameters, the size of the NMC LCP is 310 for the dual-arm system. The statistics for the NMC scheme are also summarized in the first column of Table 1 in Section 6.

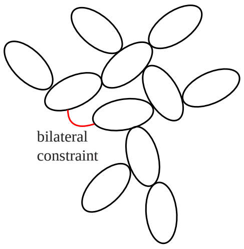

Figure 4. Tree augmented model with only loop closures modeled as bilateral constraints.

4 Minimal coordinate (MC) LCP formulation

In contrast with the NMC formulation, in the *minimal coordinates (MC)* formulation, inter-link hinges are not modeled as bilateral constraints. Instead, minimal hinge coordinates are used to parameterize the permissible hinge motion. In doing so, the number of coordinates associated with the hinge match the number of degrees of freedom for the hinge. This approach is used for all the hinges in a spanning tree for the system graph, and bilateral constraints are used only for additional loop closures that may be present in the system topology as illustrated in Figure 4.

Except for the switch from non-minimal to minimal coordinates, the development of the MC formulation largely parallels that for the NMC formulation. Hence wherever possible, we reuse the earlier notation, with the understanding that the meaning of the symbols depends on the formulation context. Thus once again, we use $\mathcal{N}$ to denote the number of degrees of freedom for the tree sub-system. With $\theta \in \mathcal{R}^{\mathcal{N}}$ denoting the vector of hinge coordinates, the minimal coordinates equations of motion for the smooth dynamics of just the tree-topology sub-system can be expressed as

$$\mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) = \mathcal{T} \tag{28}$$

where the configuration dependent matrix $\mathcal{M}(\theta) \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}$ is the *mass matrix* of the system, $\mathcal{C}(\theta, \dot{\theta}) \in \mathcal{R}^{\mathcal{N}}$ denotes the velocity dependent Coriolis and gyroscopic forces vector, and $\mathcal{T} \in \mathcal{R}^{\mathcal{N}}$ denotes the applied generalized forces. The mass matrix is symmetric and positive-definite for tree-topology systems. The configuration dependency and dense structure of $\mathcal{M}$ makes it clearly more complex than the sparse structure and constant value of the M mass matrix in the NMC formulation. On the other hand, for the dual-arm robot system in Figure 1, $\mathcal{M}$ is a compact 26-dimensional square matrix compared with the 156-dimensional square matrix M.

Let n_b denote the dimension of the bilateral constraints on the system arising from the loop closures in the system. Since n_b applies only to loop bilateral constraints, it is much smaller than n_b in the MC formulation. There exists a $G_b(\theta, t) \in \mathcal{R}^{n_b\times\mathcal{N}}$ matrix and a $\mathfrak{U}(t) \in \mathcal{R}^{n_b}$ vector that defines the velocity domain constraint equation as follows:

$$G_b(\theta, t)\dot{\theta} = \mathfrak{U}(t) \tag{29}$$

Once again we assume that $G_b(\theta, t)$ is a *full-rank* matrix.

The smooth dynamics of closed-chain systems can be obtained by modifying the tree system dynamics in Eq. 28 to include the effect of the bilateral constraints via *Lagrange multipliers*, $\lambda \in \mathcal{R}^{n_b}$, as follows

$$\begin{aligned} \mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) - G_b^*(\theta, t)\lambda &= \mathcal{T} \\ G_b(\theta, t)\dot{\theta} &= \mathfrak{U}(t) \end{aligned} \tag{30}$$

By differentiating the bilateral constraint equation Eq. 29, and including in the average force from the $p_u \in \mathcal{R}^{\mathcal{N}}$ contact impulse, Eq. 30 can be rearranged into the following descriptor form:

$$\begin{pmatrix} \mathcal{M} & -G_b^* \\ G_b & 0 \end{pmatrix} \begin{bmatrix} \ddot{\theta} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathcal{T} - \mathcal{C} + p_u/\Delta_t \\ \bar{\mathfrak{U}} \end{bmatrix} \quad \text{where} \quad \bar{\mathfrak{U}} \triangleq \dot{\mathfrak{U}}(t) - \dot{G}_b\dot{\theta} \in \mathcal{R}^{n_c} \tag{31}$$

Using the discrete Euler step approximation

$$\dot{\theta}^+ - \dot{\theta}^- = \ddot{\theta}\Delta_t \tag{32}$$

the discretized version of Eq. 31 takes the form

$$\begin{pmatrix} \mathcal{M} & -G_b^* \\ G_b & 0 \end{pmatrix} \begin{bmatrix} \dot{\theta}^+ - \dot{\theta}^- \\ p_b \end{bmatrix} = \begin{bmatrix} (\mathcal{T} - \mathcal{C})\Delta_t + p_u \\ \bar{\mathfrak{U}}\Delta_t \end{bmatrix} \quad \text{with} \quad p_b \triangleq \lambda\Delta_t \tag{33}$$

With $G_u \in \mathcal{R}^{3n_u\times\mathcal{N}}$ such that

$$\nu_u = G_u\dot{\theta} \tag{34}$$

the dual expression for the contact spatial impulses is given by

$$p_u = G_u^* F_u \overset{16}{=} G_u^* \underline{D}\underline{\beta} \tag{35}$$

Combining the complementarity conditions in Eq. 14 with Eq. 33 leads to the MC formulation version of the Eq. 9 LCP with

$$\mathfrak{M} \triangleq \left(\begin{array}{cc|cc} \mathcal{M} & -G_b^* & -G_u^*\underline{D} & 0 \\ G_b & 0 & 0 & 0 \\ \hline \underline{D}^* G_u & 0 & 0 & \hat{E} \\ 0 & 0 & -\bar{E} & 0 \end{array}\right), \quad z \triangleq \left[\begin{array}{c} \dot{\theta}^+ \\ p_b \\ \hline \underline{\beta} \\ \sigma \end{array}\right], \quad q \triangleq \left[\begin{array}{c} -\mathcal{M}\dot{\theta}^- - (\mathcal{T} - \mathcal{C})\Delta_t \\ -G_b\dot{\theta}^- - \bar{\mathfrak{U}}\Delta_t \\ \hline 0 \\ 0 \end{array}\right] \tag{36}$$

This is a mixed LCP with the top two rows correspond to equality conditions while the lower two are complementarity conditions. Its structure is very similar to the NMC formulation LCP in Eq. 27 and differs primarily in the use of minimal coordinates. The size of the MC LCP is $(\mathcal{N} + n_b + n_u(n_f + 2))$. Unlike the NMC formulation, this dimension does not depend on the number of links n. Since $\mathcal{N}$ is much smaller when using minimal coordinates, the MC LCP size is much smaller than the NMC LCP size. For the dual arm robot in Figure 1, the dimension of the MC LCP is just 50 compared with 310 for the NCP formulation.

On the other hand, evaluating $\mathfrak{M}$ for the MC LCP requires the configuration dependent and dense $\mathcal{M}$ matrix. While the composite rigid body inertia algorithm provides an efficient way to compute $\mathcal{M}$ [8], the computational complexity scales as $O(\mathcal{N}^2)$. Thus the decrease in the LCP size and solution complexity for the MC formulation are accompanied by an increase in the complexity of setting up the LCP. The cost complexity for the MC formulation is also summarized in Table 1. The solution of the MC LCP yields the new $\dot{\theta}^+$ generalized velocity value which can be integrated to propagate the θ configuration coordinates. As in the case of the NMC formulation, the bulk of the computational effort in the MC formulation is in setting up and solving the LCP problem.

5 Reduced minimal coordinate (RMC) LCP formulation

Continuing with the minimal coordinate approach, we now take further steps to reduce the size of the LCP problem. The matrix on the left of Eq. 31 can be inverted to yield the following solution for $\ddot{\theta}$:

$$\begin{aligned}
\ddot{\theta}_f &\triangleq \mathcal{M}^{-1}\left[\mathcal{T} - \mathcal{C} + p_u/\Delta_t\right] && \text{(37a)}\\
\lambda &= \left[G_b\mathcal{M}^{-1}G_b^*\right]^{-1}\left(-G_b\ddot{\theta}_f + \bar{\mathfrak{U}}\right) && \text{(37b)}\\
\ddot{\theta} &= \ddot{\theta}_f + \mathcal{M}^{-1}G_b^*\,\lambda\\
&\overset{37b}{=} \left[I - \mathcal{M}^{-1}G_b^*\left[G_b\mathcal{M}^{-1}G_b^*\right]^{-1}G_b\right]\ddot{\theta}_f + \mathcal{M}^{-1}G_b^*\left[G_b\mathcal{M}^{-1}G_b^*\right]^{-1}\bar{\mathfrak{U}} && \text{(37c)}
\end{aligned}$$

Using Eq. 32, we obtain

$$\begin{aligned}
\dot{\theta}^+ &\overset{32}{=} \dot{\theta}^- + \ddot{\theta}\Delta_t\\
&\overset{37c}{=} \dot{\theta}^- + \left[I - \mathcal{M}^{-1}G_b^*\left[G_b\mathcal{M}^{-1}G_b^*\right]^{-1}G_b\right]\Delta_t\ddot{\theta}_f + \mathcal{M}^{-1}G_b^*\left[G_b\mathcal{M}^{-1}G_b^*\right]^{-1}\Delta_t\bar{\mathfrak{U}} \qquad \text{(38)}\\
&\overset{37a}{=} Yp_u + X
\end{aligned}$$

where

$$\begin{aligned}
Y &\triangleq \mathcal{M}^{-1} - \mathcal{M}^{-1}G_b^*(G_b\mathcal{M}^{-1}G_b^*)^{-1}G_b\mathcal{M}^{-1} \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}\\
\text{and}\quad X &\triangleq \dot{\theta}^- + Y(\mathcal{T} - \mathcal{C})\Delta_t + \mathcal{M}^{-1}G_b^*\left[G_b\mathcal{M}^{-1}G_b^*\right]^{-1}\bar{\mathfrak{U}}\Delta_t \in \mathcal{R}^{\mathcal{N}}
\end{aligned} \qquad \text{(39)}$$

Thus

$$\underline{D}^*v_u^+ \overset{34}{=} \underline{D}^*G_u\dot{\theta}^+ \overset{35,38}{=} \underline{D}^*G_uY\,G_u^*\underline{D}\,\underline{\beta} + \underline{D}^*G_uX \qquad \text{(40)}$$

Using this allows us to eliminate $\dot{\theta}^+$ and p_b from the MC LCP formulation in Eq. 36 to obtain the following *Reduced Minimal Coordinate (RMC)* formulation LCP:

$$\mathfrak{M} \triangleq \begin{pmatrix} \underline{D}^*G_uY\,G_u^*\underline{D} & \hat{E}\\ \bar{E} & 0 \end{pmatrix}, \quad z \triangleq \begin{bmatrix} \underline{\beta}\\ \sigma \end{bmatrix}, \quad q \triangleq \begin{bmatrix} \underline{D}^*G_uX\\ 0 \end{bmatrix} \qquad \text{(41)}$$

Since there are no equality conditions, this is a standard rather than a mixed LCP. The size of this RMC LCP is $n_u(n_f+2)$. It is notable that the size of the LCP does not depend on the number of links n, the number of degrees of freedom $\mathcal{N}$, nor the n_b dimension of the bilateral constraints. It only depends on the number of contact constraint nodes. Thus the dimension of this LCP is even smaller than that for the MC formulation. For the dual arm robot system, the dimension of the LCP is 24. On the other hand, computing $\mathfrak{M}$ for the RMC LCP requires the Y matrix in Eq. 39, which requires the configuration dependent $\mathcal{M}^{-1}$ matrix and several expensive matrix/matrix products. These computations are of $O(\mathcal{N}^3)$ computational complexity. Once again, while the RMC formulation has successfully reduced the LCP size and consequently its solution complexity, this reduction has been accompanied by a significant increase in the complexity of setting up the LCP problem. The cost complexity for the RMC formulation is summarized in Table 1.

In contrast with the NMC and MC formulations, the solution of the RMC LCP does not by itself yield the new system velocity or state. Instead the following sequence of steps is needed to obtain the new state values:

1. Assemble and solve the RMC LCP in Eq. 41 to obtain $\underline{\beta}$ and σ. Use $\underline{\beta}$ in Eq. 35 to obtain the p_u contact impulse vector.

2. Use p_u in Eq. 38 to compute the new $\dot{\theta}^+$ system velocity. This can be integrated to obtain the new system configuration coordinates θ.

Thus, the RMC LCP by itself does not do all the work, and the additional step (2) is needed to complete the computation of the new $\dot{\theta}^+$ system velocity coordinates.

The formulation developed by Trinkle [2] is a hybrid combination of the NMC and RMC formulations. Trinkle's setup allows the use of general coordinates for describing the smooth equations of motion. However, instead of eliminating the hinge bilateral constraints by using minimal hinge coordinates a pair of symmetric (positive and negative) complementarity conditions are added to enforce the equality condition for each hinge constraint. This inflates the size of the LCP much like the NMC approach. However, Trinkle;s approach is similar to the RMC in eliminating the velocity coordinates and the loop closure bilateral constraint Lagrange multipliers from the LCP problem to obtain an LCP similar in form to Eq. 41.

6 Operational space (OS) LCP formulation

So far we have found that the reductions in LCP size have had the side-effect of increasing the LCP setup complexity. In this section we look into reducing such setup complexity using low-order articulated system dynamics algorithms. From Eq. 31 we have

$$\begin{aligned} 0 &\overset{29}{=} G_b\ddot{\theta} - \overline{\mathfrak{U}} \\ &\overset{31}{=} G_b\mathcal{M}^{-1}\left[\mathcal{T} - \mathcal{C} + G_b^*\lambda + p_u/\Delta_t\right] - \overline{\mathfrak{U}} \\ &= G_b\mathcal{M}^{-1}G_b^*\lambda + G_b\mathcal{M}^{-1}p_u/\Delta_t + \alpha_b^f \quad \text{where} \quad \alpha_b^f \triangleq G_b\mathcal{M}^{-1}(\mathcal{T}-\mathcal{C}) - \overline{\mathfrak{U}} \\ &\overset{35}{=} G_b\mathcal{M}^{-1}G_b^*\lambda + G_b\mathcal{M}^{-1}G_u^*\underline{D}\underline{\beta}/\Delta_t + \alpha_b^f \end{aligned} \tag{42}$$

The above expression characterizes the equality condition on the dynamics from the bilateral constraints. The relative linear acceleration of the contact nodes is obtained by differentiating Eq. 34 to obtain

$$\dot{v}_u = G_u\ddot{\theta} + \dot{G}_u\dot{\theta} \overset{31}{=} G_u\mathcal{M}^{-1}\left[\mathcal{T} - \mathcal{C} + G_b^*\lambda + p_u/\Delta_t\right] + \dot{G}_u\dot{\theta} \tag{43}$$

The discretized approximation $(v_u^+ - v_u^-) = \dot{v}_u\Delta_t$ of this equation leads to

$$v_u^+ \overset{35,\,43}{=} G_u\mathcal{M}^{-1}G_b^*\lambda\Delta_t + G_u\mathcal{M}^{-1}G_u^*\underline{D}\,\underline{\beta} + v_u^- + \alpha_u^f\Delta_t \quad \text{where} \quad \alpha_u^f \triangleq G_u\mathcal{M}^{-1}(\mathcal{T}-\mathcal{C}) + \dot{G}_u\dot{\theta} \tag{44}$$

Combining the complementarity conditions in Eq. 14 with Eq. 42 and Eq. 44 yields the following mixed LCP for the system:

$$\mathfrak{M} \triangleq \left(\begin{array}{cc|c} G_b\mathcal{M}^{-1}G_b^* & G_b\mathcal{M}^{-1}G_u^*\underline{D} & 0 \\ \hline \underline{D}^*G_u\mathcal{M}^{-1}G_b^* & \underline{D}^*G_u\mathcal{M}^{-1}G_u^*\underline{D} & \hat{E} \\ 0 & \bar{E} & 0 \end{array}\right), \quad z \triangleq \left[\begin{array}{c} p_b \\ \hline \underline{\beta} \\ \sigma \end{array}\right], \quad q \triangleq \left[\begin{array}{c} \alpha_b^f\Delta_t \\ \hline \underline{D}^*(v_u^- + \alpha_u^f\Delta_t) \\ 0 \end{array}\right] \tag{45}$$

This $\mathfrak{M}$ matrix still requires the configuration dependent $\mathcal{M}^{-1}$ matrix whose evaluation if of $O(\mathcal{N}^3)$ computational complexity. We next look more closely at the structure of the G_u and G_b matrices.

The unilateral and bilateral constraints are associated with nodes on the bodies. Let us denote the number of this overall set of nodes involved in the unilateral and bilateral constraints as n_c. Denoting the spatial velocities of these nodes by the stacked vector $\mathcal{V}_c \in \mathcal{R}^{6n_c}$, there exist matrices $\mathcal{Q}_u \in \mathcal{R}^{3n_u \times 6n_c}$ and $\mathcal{Q}_b \in \mathcal{R}^{n_b \times 6n_c}$ such that the unilateral and bilateral velocity constraint equations can be expressed as[3]

$$v_u = \mathcal{Q}_u\mathcal{V}_c \quad \text{and} \quad \mathcal{Q}_b\mathcal{V}_c = \mathfrak{U} \tag{46}$$

Let $\mathcal{J} \in \mathcal{R}^{6n_c \times \mathcal{N}}$ denote the Jacobian for the constraint nodes, so that

$$\mathcal{V}_c = \mathcal{J}\dot{\theta} \tag{47}$$

[3] $\mathcal{Q}_u$ has the same structure as the $\mathcal{Q}_b$ constraint mapping matrix for bilateral constraints for three degree of freedom spherical hinges.

It follows from Eq. 29, Eq. 34, Eq. 46 and Eq. 47 that G_u and G_b have the following form:

$$G_u = \mathcal{Q}_u \mathcal{J} \quad \text{and} \quad G_b = \mathcal{Q}_b \mathcal{J} \tag{48}$$

With

$$\underline{\Lambda} \triangleq \mathcal{J}\mathcal{M}^{-1}\mathcal{J}^* \in \mathcal{R}^{6n_c \times 6n_c} \tag{49}$$

we can use Eq. 48 to re-express $\mathfrak{M}$ in Eq. 45 as

$$\mathfrak{M} = \begin{pmatrix} \mathcal{Q}_b \underline{\Lambda} \mathcal{Q}_b^* & \mathcal{Q}_b \underline{\Lambda} \mathcal{Q}_u^* \underline{D} & 0 \\ \underline{D}^* \mathcal{Q}_u \underline{\Lambda} \mathcal{Q}_b^* & \underline{D}^* \mathcal{Q}_u \underline{\Lambda} \mathcal{Q}_u^* \underline{D} & \hat{E} \\ 0 & \overline{E} & 0 \end{pmatrix} = \left(\begin{array}{c|c} \begin{bmatrix} \mathcal{Q}_b \\ \underline{D}^* \mathcal{Q}_u \end{bmatrix} \underline{\Lambda} \left[\mathcal{Q}_b^*, \; \mathcal{Q}_u^* \underline{D} \right] & \begin{matrix} 0 \\ \hat{E} \end{matrix} \\ \hline 0 \qquad\qquad \overline{E} & 0 \end{array} \right) \tag{50}$$

The $\underline{\Lambda} = \mathcal{J}\mathcal{M}^{-1}\mathcal{J}^*$ matrix definition in Eq. 49 is precisely the mathematical expression for the inverse of the *operational space inertia* matrix that is used in the operational space approach for robot manipulation and control [12, 13]. Based on this structural similarity, we borrow and extend the operational space terminology to our current context with the constraint nodes forming the operational space nodes. Also, borrowing terminology, we refer to $\underline{\Lambda}$ as the *operational space compliance matrix (OSCM)* matrix. The invertibility of $\underline{\Lambda}$ does not depend on $\mathcal{J}$ being invertible – only that $\mathcal{J}$ have full row-rank. When it exists, the inverse of $\underline{\Lambda}$ is referred to as the *operational space inertia*. The properties of the OSCM are discussed in detail in [10].

The property of the $\underline{\Lambda}$ matrix that is of importance for us is the availability of algorithms of $O(\mathcal{N}) + O(n_c^2)$ computational complexity for evaluating $\underline{\Lambda}$ [10, 14]. The low-order of these algorithms is remarkable given the presence of $\mathcal{M}^{-1}$ in the expression for $\underline{\Lambda}$, since evaluating $\mathcal{M}$ and $\mathcal{M}^{-1}$ individually require $O(\mathcal{N}^2)$ and $O(\mathcal{N}^3)$ computations respectively. This algorithm reduces the complexity of evaluating $\mathfrak{M}$ in Eq. 50 from $O(\mathcal{N}^3)$ to the much smaller $O(\mathcal{N}) + O(n_c^2)$ computational complexity. The low complexity algorithm for evaluating $\underline{\Lambda}$ is based on an analytical transformation of Eq. 49, followed by a disjoint decomposition of the matrix into block diagonal, and upper and lower triangular components that can be computed recursively. A summary of this structure-based analysis and accompanying algorithms using spatial operator techniques is described in the appendix. An alternative sparsity based technique for evaluating $\underline{\Lambda}$ is described in reference [15].

With $\ddot{\theta}_f \triangleq \mathcal{M}^{-1}(\mathcal{T} - \mathcal{C})$,

$$\begin{aligned} \alpha_b^f &\overset{42,48}{=} \mathcal{Q}_b \mathcal{J} \ddot{\theta}_f - \overline{\mathfrak{U}} \overset{31,47}{=} \mathcal{Q}_b (\mathcal{J}\ddot{\theta}_f + \dot{\mathcal{J}}\dot{\theta}) + \dot{\mathcal{Q}}_b \mathcal{V}_c - \dot{\mathfrak{U}} = \mathcal{Q}_b \alpha^f + \dot{\mathcal{Q}}_b \mathcal{V}_c - \dot{\mathfrak{U}} \\ \alpha_u^f &\overset{44,48}{=} \mathcal{Q}_u \mathcal{J} \ddot{\theta}_f + \dot{G}_u \overset{47}{=} \mathcal{Q}_u (\mathcal{J}\ddot{\theta}_f + \dot{\mathcal{J}}\dot{\theta}) + \dot{\mathcal{Q}}_u \mathcal{V}_c = \mathcal{Q}_u \alpha^f + \dot{\mathcal{Q}}_u \mathcal{V}_c \end{aligned} \tag{51}$$

where

$$\alpha^f = \mathcal{J}\ddot{\theta}_f + \dot{\mathcal{J}}\dot{\theta} \tag{52}$$

Physically, α^f is the stacked vector of spatial accelerations of the constraint nodes in the absence of the bilateral and contact constraints.

Using Eq. 50 and Eq. 51, the Eq. 45 LCP can be re-expressed as the following *Operational Space (OS)* formulation LCP:

$$\mathfrak{M} \triangleq \left(\begin{array}{c|c} \begin{bmatrix} \mathcal{Q}_b \\ \underline{D}^* \mathcal{Q}_u \end{bmatrix} \underline{\Lambda} \left[\mathcal{Q}_b^*, \; \mathcal{Q}_u^* \underline{D} \right] & \begin{matrix} 0 \\ \hat{E} \end{matrix} \\ \hline 0 \qquad\qquad \overline{E} & 0 \end{array} \right), \; z \triangleq \left[\begin{array}{c} p_b \\ \underline{\beta} \\ \hline \sigma \end{array} \right], \; q \triangleq \left(\begin{array}{c} \begin{bmatrix} \mathcal{Q}_b \\ \underline{D}^* \mathcal{Q}_u \end{bmatrix} \Delta_t \alpha^f + \begin{bmatrix} \dot{\mathcal{Q}}_b \\ \underline{D}^* \dot{\mathcal{Q}}_u \end{bmatrix} \Delta_t \mathcal{V}_c + \begin{bmatrix} -\dot{\mathfrak{U}} \\ \underline{D}^* \nu_u^- \end{bmatrix} \\ \hline 0 \end{array} \right) \tag{53}$$

This is a mixed LCP, with the first row corresponding to an equality condition while the bottom two rows correspond to complementarity conditions. The size of this LCP is $(n_b + n_u(n_f + 2))$. Like the RMC formulation, the size of this LCP does not depend on the number of links n or the number of degrees of freedom $\mathcal{N}$, but it does depend on the n_b dimension of the loop closure bilateral constraints. The dimension of the OS LCP is moderately larger than the RMC LCP but smaller than the MC LCP. Typically, $\dot{\mathcal{Q}}_u$, $\dot{\mathcal{Q}}_b$ and $\dot{\mathfrak{U}}$ are all zero leading to a simpler q in Eq. 53. For the dual arm robot system, the dimension of the OS LCP is 24.

Computing $\mathfrak{M}$ for the OS LCP requires the configuration dependent $\underline{\Lambda}$ matrix Eq. 53 whose evaluation is of $O(\mathcal{N}) + O(n_c^2)$ computational complexity which is much smaller than the $O(\mathcal{N}^3)$ complexity for evaluating $\mathfrak{M}$ for the RMC method. Thus in comparison with the RMC formulation, while the OS formulation increases the size of the LCP by a modest n_b, it drastically reduces the LCP setup complexity. The result is a significant reduction in the overall complexity of the contact dynamics computations for the OS formulation.

Like the RMC formulations, the solution of the LCP does not by itself yield the new system velocity or state. Instead the following sequence of steps is needed to obtain the new state values:

1. Assemble and solve the OS LCP in Eq. 53 to obtain $\mathfrak{p}_b$, $\underline{\beta}$ and σ. Use $\underline{\beta}$ in Eq. 35 to obtain the $\mathfrak{p}_u$ contact impulse vector.

2. Use $\mathfrak{p}_b$ and $\mathfrak{p}_u$ in Eq. 33 to obtain the new $\dot{\theta}^+$ system velocity using the $O(\mathcal{N})$ articulated body forward dynamics algorithm. This can be integrated to obtain the new system configuration coordinates θ.

Like the RMC formulation, the LCP by itself does not do all the work in the OS formulation, but instead the additional step (2) is needed to complete the computation of the new $\dot{\theta}^+$ system velocity coordinates.

	LCP Formulation			
Property	**NMC**	**MC**	**RMC**	**OS**
Coordinates type	Non-minimal	Minimal	Minimal	Minimal
LCP assembly complexity	O(n)	$O(\mathcal{N}^2)$	$O(\mathcal{N}^3)$	$O(\mathcal{N}) + O(n_c^2)$
LCP dimension	$6n + n_b + n_u(n_f + 2)$	$\mathcal{N} + n_b + n_u(n_f + 2)$	$n_u(n_f + 2)$	$n_b + n_u(n_f + 2)$
Dual-arm LCP dimension	310	50	24	24
Ancillary dynamics steps	None	None	Evaluate $\mathfrak{p}_u$ and $\dot{\theta}^+$	Evaluate $\mathfrak{p}_u$ and $\dot{\theta}^+$

Table 1. *A comparison of the features of the different NMC, MC, RMC and OS formulations for contact and collision dynamics. The LCP dimension size is for the reference dual-arm robot problem, while the LCP assembly complexity highlights just the major contributors.*

The LCP formulation developed by Yamane and Nakamura [7] makes use of the *divide and conquer algorithm (DCA)* [16] techniques and is a special case of the OS formulation. The OS formulation here is however more general since it handles loop closure bilateral constraints, exploits operational space techniques to reduce computational complixity, and as described later, handles collision dynamics.

Table 1 summarizes the dimensions and computational compleixty for all the formulations discussed so far. The trend across the NMC, MC and RMC formulations is that the reduction in the size of the LCP shifts costs to the LCP setup process. While the initial form of the OS formulation LCP in Eq. 45 also follows this trend, the restructured Eq. 53 LCP breaks the pattern by restructuring the LCP to take advantage of low-order, structure-based algorithms for articulated system dynamics.

7 Collision dynamics

In this section we develop extensions to the OS LCP formulation for handling the dynamics of collision events. During inelastic collisions some of the impact energy is lost. The *coefficient of restitution*, $\epsilon(i)$ defines the fraction that remains after a collision. The complementarity approach to modeling collisions breaks up the collision event into a pair of instantaneous *compression* and *decompression* phases [3]. During the compression phase, the collision impulse is stored, and during decompression, a fraction of the collision impulse is recovered. We will make use of time discretized equations with impulses developed for contact dynamics, but with $\Delta_t = 0$ since collision events are instantaneous.

7.1 Compression

At the i^{th} contact undergoing collision, the compression phase is instantaneous and impulsively changes the relative linear contact velocity from $v_u^-(i)$ to a new $v_c^+(i)$ value with a non-negative normal component. The compression impulse is denoted $\mathfrak{p}_c(i)$. The mixed LCP problem for the compression phase is obtained by setting $\Delta_t = 0$ in Eq. 53 to obtain

$$w = \mathfrak{M}z + q \perp z \quad \text{with} \quad q \overset{\triangle}{=} \begin{pmatrix} \begin{bmatrix} -\dot{\mathfrak{U}} \\ \underline{D}^* v_u^- \end{bmatrix} \\ 0 \end{pmatrix} \tag{54}$$

The LCP solution is used to instantaneously (i.e. impulsively) propagate the state for the compression phase as follows:

$$\begin{aligned}
\mathrm{p}_c &= \mathcal{Q}_u^* \underline{D}\,\underline{\beta} + \mathcal{Q}_b^* \mathrm{p}_b \\
\dot{\theta}^c &= \dot{\theta}^- + \mathcal{M}^{-1} \mathcal{J}^* \mathrm{p}_c \\
v_c^+ &= \mathcal{J} \dot{\theta}^c
\end{aligned} \tag{55}$$

7.2 Decompression

The decompression phase applies an additional impulse of magnitude $\epsilon(i)[0,\ \hat{n}^*(i)p_c(i)]$ for the i^{th} contact along the normal from the impulse stored during the compression phase. The recovered ϑ decompression impulse is

$$\vartheta \ \triangleq\ \mathrm{col}\{(\epsilon(i)[0,\ \hat{n}^*(i)]p_c(i))\,\hat{n}(i)\}_{i=1}^{n_u} \quad \in \mathcal{R}^{3n_u} \tag{56}$$

The decompression LCP is obtained by updating Eq. 42 and Eq. 43 to include the additional ϑ impulse. This leads to a decompression LCP problem that is the mixed LCP in Eq. 53 with $\Delta_t = 0$, the contact linear velocity v_u^- replaced with v_c^+, and an additional $\begin{bmatrix} \mathcal{Q}_b \\ \underline{D}^* \mathcal{Q}_u \end{bmatrix} \underline{\Lambda} \mathcal{Q}_u^* \vartheta$ term for the recovered impulse included in the q LCP vector term. The resulting decompression phase LCP is

$$w = \mathfrak{M} z + q \quad \perp \quad z \quad \text{with} \quad q \ \triangleq \begin{pmatrix} \begin{bmatrix} -\dot{\mathfrak{U}} \\ \underline{D}^* v_c^+ \end{bmatrix} + \begin{bmatrix} \mathcal{Q}_b \\ \underline{D}^* \mathcal{Q}_u \end{bmatrix} \underline{\Lambda} \mathcal{Q}_u^* \vartheta \\ 0 \end{pmatrix} \tag{57}$$

The LCP solution for the decompression impulse can include additional contact impulse terms that ensure that the normal component of the relative linear velocity at the end of the decompression step remains non-negative. The LCP solution is used to instantaneously propagate the state for the decompression phase as follows:

$$\begin{aligned}
\mathrm{p} &= \mathcal{Q}_u^* \underline{D}\,\underline{\beta} + \mathcal{Q}_b^* \lambda + \mathcal{Q}_u^* \vartheta \\
\dot{\theta}^+ &= \dot{\theta}^c + \mathcal{M}^{-1} \mathcal{J}^* \mathrm{p}
\end{aligned} \tag{58}$$

When $\epsilon(i) = 0$, the collision is completely inelastic, and there is no decompression phase. However, in general, each collision event requires the solution of two LCP's in this approach.

8 Simulation results

We use a simulation of a multi-link pendulum colliding with itself and the environment to quantitatively evaluate the performance of the OS formulation. This example also allows us to paramaterically measure the performance improvement as a function of the problem dimension by varying the number of links in the pendulum. The environment consists of a floor and a wall located 4m away. The multi-link pendulum consists of n identical 1kg mass spherical bodies connected with pin hinges. The radius of the sphere is scaled based on the number of links to maintain a 12m overall length of the pendulum. The pendulum base is located at a height of 10m. The open source Bullet software [17] is used for collision detection, and the PATH software [18] for solving mixed complementarity problems. The simulation uses a time step of 0.1ms, with a 0.5 coefficient of friction and a 0.7 coefficient of restitution to simulate inelastic collisions. The pendulum starts at an angle of $\pi/4$ radians with an initial angular velocity of 1 radian/s and a gravitational acceleration of 9.8m/s^2.

As the pendulum swings from left to right, it collides with the ground, bounces off of the ground, and eventually collides with the wall on the right. In the course of the sequence, multiple links are at times in collision with the ground, the wall and with each other. Figure 5 contains a sequence of screen shots from such a simulation for a 12-link pendulum. A video of the simulation is included in the media clips accompanying this article. We have simulated this contact and collision dynamics scenario using two different techniques. The first technique is the minimal coordinate OS formulation described in Section 6.

The second technique, that we refer to as the *NMC/OS formulation*, is a non-minimal coordinate variant of the OS formulation. Similar to the NMC method, each link is treated as an independent body, and the hinges are handled as bilateral constraints between the neighboring links with $n_b = 6n - \mathcal{N}$. The NMC/OS LCP has the same form as the OS LCP in Eq. 53, except that the OSCM is the non-minimal coordinate $\underline{\Lambda} = \mathcal{J} M^{-1} \mathcal{J}^*$, instead of Eq. 49. The NMC/OS $\underline{\Lambda}$ is a much larger matrix but with a simple block diagonal structure. This NMC/OS LCP does not include system velocity coordinates $\mathcal{V}$ in z and thus is smaller than the NMC LCP.

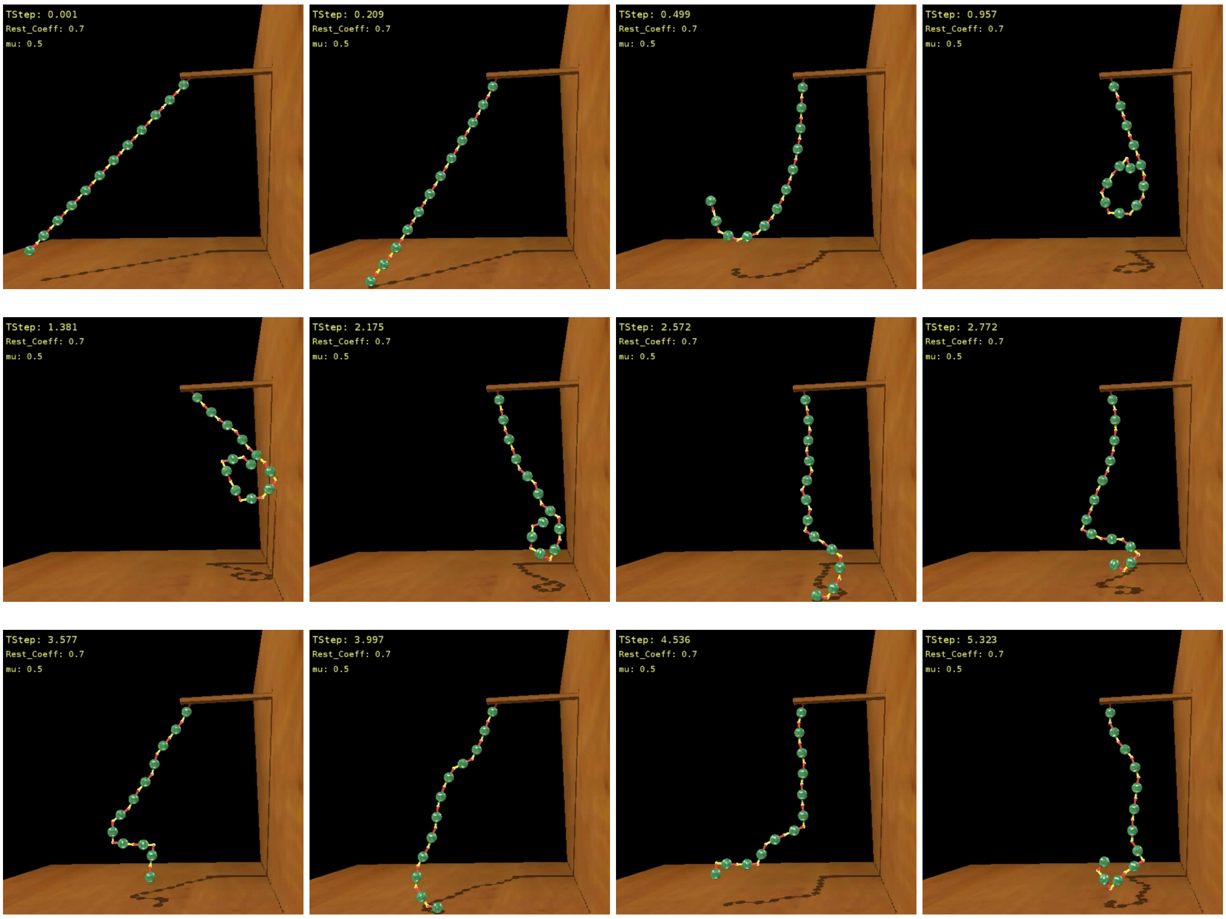

Figure 5. Time series capture of swinging pendulum simulation with 12 links

The simulation results from the two methods show good agreement. Figure 6 shows example plots of the height and normal velocity of the last link of the 12-body pendulum from the two simulation methods. The vertical spikes in the velocity plot are discontinuous jumps from collisions involving the pendulum bodies. The small trajectory differences in the plots decrease further when the time step size is reduced.

Table 2 compares the computational complexity of the OS and the NMC/OS formulations for pendulums with the number of links varying between 3 and 30 links. The table also lists the LCP size for the OS, NMC/OS and the NMC formulations. The size of the LCP remains a constant value of 24 for the standard OS formulation even when the number

	LCP size			Computation Time (s)		
Number of links	**OS**	**NMC/OS**	**NMC**	**OS**	**NMC/OS**	**Speed up**
3	24	39	57	13.33	47.79	3.6
6	24	54	90	10.50	68.71	6.5
12	24	84	156	19.06	305.28	16.1
15	24	99	189	21.76	558.10	25.6
24	24	144	288	38.20	1899.86	49.7
30	24	174	354	73.74	4100.51	55.6

Table 2. *A comparison of the LCP size and computational time for the OS and NMC/OS formulations for the multi-link pendulum example with different number of links. The LCP size assumes 4 contacts and $n_f = 4$. The three LCP size columns are for the OS, the NMC/OS and the NMC formulations. The speed up value is the ratio of the NMC/OS to the OS formulation simulation times.*

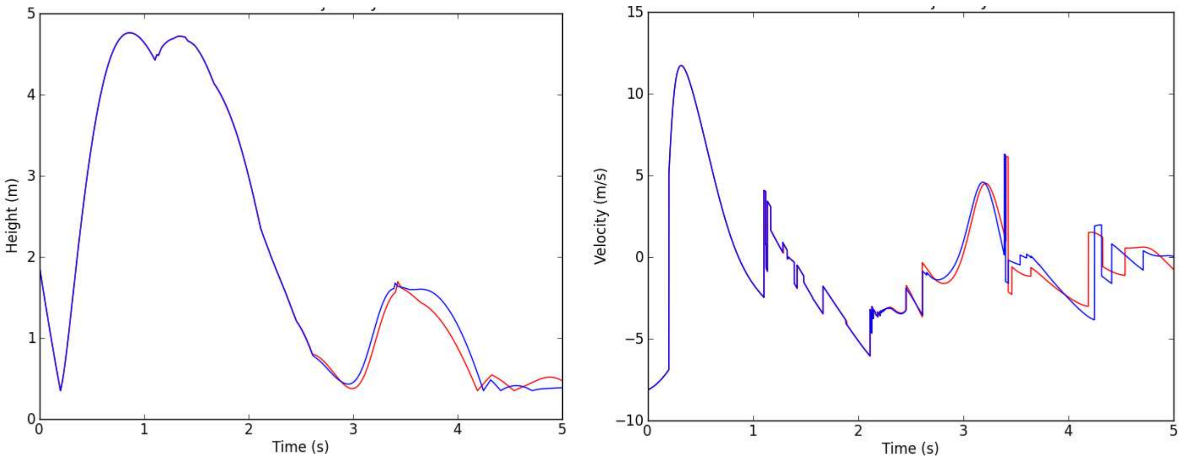

Figure 6. *Comparisons of the height and normal velocity of the last link using the OS (red) and NMC/OS (blue) formulation based simulations for a 12-body pendulum.*

of links and degrees of freedom in the system is increased. In contrast, the LCP size increases with the increase in the number of links and degrees of freedom for the NMC/OS and the NMC formulations. We also observe that the OS method is about 3.6 times faster for the 3 link pendulum case, and over 55 times faster for the 30 link pendulum when compared with the the NMC/OS method. The performance gap widens substantially as the number of links in the system is increased. The performance gap between the OS and NMC formulations will be even greater due to the even larger size of the NMC LCP.

We have also applied the OS formulation to simulate manipulation tasks for the dual arm system in Figure 1. Video clips showing the dual arm removal of a wheel from its hub, and the grasping and hand off of an impact driver from one hand to the other in simulation are included in the accompanying media clips. In each of these cases, the contact and collision dynamics events during grasping and other interactions are simulated using the OS formulation. Unlike the serial-chain structure of the pendulum system, the dual-arm system has a more general tree-topology.

9 Conclusions

In this article we have described a progression of formulations for the contact and collision dynamics of multi-link articulated systems with the goal of reducing computational complexity. Along the way, we have clarified the relationships among the different approaches and those in the literature. Our strategy has been to find a formulation that best exploits the available low-order articulated body dynamics algorithms to reduce the overall computational complexity.

The formulations studied here vary in the size of the LCP, the complexity of setting up the LCP, and the ancillary dynamics steps needed to complete the dynamics solution. The generally observed trend is that the reduction in the LCP size shifts the computational burden from solving the LCP problem, to the setting up of the LCP problem. The widely used NMC non-minimal coordinate formulation is the simplest and cheapest to set up, but also the most expensive to solve due to its large dimension. The RMC minimal coordinate approach on the other hand has the smallest LCP dimension, but one that is the most expensive to set up. In the RMC approach, the size of the LCP problem in is just $(n_u(n_f+2)+n_b)$, which is independent of the number of links, the number of degrees of freedom and the dimension of the bilateral constraints on the system. In contrast, the size of the corresponding NMC LCP is larger by $6n - \mathcal{N}$. For a 6-link manipulator with 6 degrees of freedom, this amounts to difference in dimension of 30.

The OS formulation shares the small LCP dimension property of the RMC approach, with an LCP that is larger by only the modest dimension of the loop closure bilateral constraints, n_b. We show that the OS formulation can be restructured in a form that expresses its LCP matrix in terms of the operational space OSCM matrix for the constraint nodes. This insight allows us to apply low-order, structure-based computational algorithms available for the OSCM to significantly reduce the complexity of setting up the OS LCP. Consequently the OS formulation is the lowest overall complexity formulation with a small LCP along with low complexity algorithms for setting up the LCP. Focusing on this option, we describe extensions of the contact dynamics formulation to handle elastic and inelastic collision dynamics. The OS formulation's use of minimal coordinates also results in the automatic enforcement of the inter-link hinge bilateral constraints and avoids the need for additional bilateral constraint error control schemes. The benchmark simulations using a pendulum system show a widening performance improvement using the OS formulation as the number of bodies is increased. For the 30 link pendulum system, the OS formulation is over 55 times faster than the NMC/OS approach. An area of future work is the integration of the OS formulation with the large variety of time stepping schemes that are in development for increasing the robustness and accuracy of contact and collision non-smooth dynamics [19].

Acknowledgments

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 and funded through the internal Research and Technology Development program.[4]

References

[1] D. Stewart and Jeffrey C Trinkle. An implicit time-stepping scheme for rigid body dynamics with Coulomb friction. In Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), pages 162–169. Ieee, 2000.

[2] Jeffrey C Trinkle. Formulation of Multibody Dynamics as Complementarity Problems. In ASME International Design Engineering Technical Conference, Chicago, IL, September 2003.

[3] Mihai Anitescu and F A Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. Nonlinear Dynamics, 14(3):231–247, 1997.

[4] F Pfeiffer. Mechanical System Dynamics. Springer, 2005.

[5] Abhinandan Jain, Cory Crean, Calvin Kuo, Hubertus von Bremen, and Steven Myint. Minimal Coordinate Formulation of Contact Dynamics in Operational Space. In Robotics Science and Systems, Sydney, Australia, 2012.

[6] Alessandro Tasora and Mihai Anitescu. A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics. Computer Methods in Applied Mechanics and Engineering, 200(5-8):439–453, January 2011.

[7] Katsu Yamane and Yoshihiko Nakamura. A Numerically Robust LCP Solver for Simulating Articulated Rigid Bodies in Contact. In Robotics: Science and Systems IV, chapter A numerica, pages 89–104. MIT Press, 2009.

[8] M W Walker and David E Orin. Efficient Dynamic Computer Simulation of Robotic Mechanisms. ASME Journal of Dynamic Systems, Measurement, and Control, 104(3):205–211, September 1982.

[9] Roy Featherstone. Rigid Body Dynamics Algorithms. Springer Verlag, 2008.

[10] Abhinandan Jain. Robot and Multibody Dynamics: Analysis and Algorithms. Springer, 2011.

[11] Abhinandan Jain, Cory Crean, Calvin Kuo, and Marco B. Quadrelli. Efficient Constraint Modeling for Closed-Chain Dynamics. In The 2nd Joint International Conference on Multibody System Dynamics, Stuttgart, Germany, 2012.

[12] Oussama Khatib. Object Manipulation in a Multi-Effector System. In 4th International Symposium on Robotics Research, pages 137–144, Santa Cruz, CA, May 1988.

[13] Oussama Khatib. A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. IEEE Journal of Robotics and Automation, RA-3(1):43–53, February 1987.

[14] K Kreutz-Delgado, Abhinandan Jain, and Guillermo Rodriguez. Recursive formulation of operational space control. Internat. J. Robotics Res., 11(4):320–328, 1992.

[15] Roy Featherstone. Exploiting Sparsity in Operational-space Dynamics. Internat. J. Robotics Res., 29(1992):1353–1368, September 2010.

[16] Roy Featherstone. A divide-and-conquer articulated-body algorithm for parallel O(log(n)) calculation of rigid-body dynamics. Part 2: Basic algorithm. Internat. J. Robotics Res., 18(9):867˜875, 1999.

[4] ©2013 California Institute of Technology. Government sponsorship acknowledged.

[17] Bullet Physics Library, 2013.

[18] The PATH Solver, 2012.

[19] Christian Walter Studer. Augmented time-stepping integration of non-smooth dynamical systems. PhD thesis, ETH Zurich, 2008.

[20] Abhinandan Jain. Graph Theoretic Foundations of Multibody Dynamics Part II: Analysis and Algorithms. Multibody System Dynamics, 26(3):335–365, October 2011.

[21] Guillermo Rodriguez, Abhinandan Jain, and K Kreutz-Delgado. A spatial operator algebra for manipulator modeling and control. Internat. J. Robotics Res., 10(4):371, 1991.

10 Appendix

The operational space for the multi-link system is defined by the configuration of the set of constraint nodes on the system. The key implementation and computational challenge for setting up the OS formulation LCP in Eq. 53 is the need for evaluating the $\underline{\Lambda}$ matrix. As seen in Eq. 49, $\underline{\Lambda}$ involves the configuration dependent matrix products of the Jacobian matrix and the mass matrix inverse. A direct evaluation of this expression requires $O(\mathcal{N}^3)$ computations. However references [10, 14, 20] have used spatial operators to develop simpler and recursive computational algorithms for $\underline{\Lambda}$ that are of only $O(\mathcal{N})$ complexity. We briefly describe the underlying analysis and structure of this algorithm, and refer the reader to [10, 14, 20] for notation and derivation details.

10.1 Spatial operator factorization of $\mathcal{M}^{-1}$

We begin with the following key spatial operator based analytical results that provide explicit, closed-form expressions for the factorization and inversion of a tree mass matrix [10, 21]:

$$\begin{aligned}
\mathcal{M} &= H\phi M\phi^* H^* \\
\mathcal{M} &= [I + H\phi\mathcal{K}]\,\mathcal{D}\,[I + H\phi\mathcal{K}]^* \\
[I + H\phi\mathcal{K}]^{-1} &= [I - H\psi\mathcal{K}] \\
\mathcal{M}^{-1} &= [I - H\psi\mathcal{K}]^*\,\mathcal{D}^{-1}\,[I - H\psi\mathcal{K}]
\end{aligned} \tag{59}$$

The first expression defines the Newton-Euler operator factorization of the mass matrix $\mathcal{M}$ in terms of the H hinge articulation, the ϕ rigid body propagation and the M link spatial inertia operators. While this factorization has non-square factors, the second expression describes an alternative factorization involving only square factors with block diagonal $\mathcal{D}$ and block lower-triangular $[I + H\phi\mathcal{K}]$ matrices. This factorization involves new spatial operators that are associated with the *articulated body (AB)* forward dynamics algorithm [9, 20] for the system. The next expression describes an analytical expression for the inverse of the $[I + H\phi\mathcal{K}]$ operator. Using this leads to the final analytical expression for the inverse of the mass matrix. These operator expressions hold generally for tree-topology systems irrespective of the number of bodies, the types of hinges, the specific topological structure, and even for non-rigid links [10].

10.2 The Ω extended operational space compliance matrix

With $\mathcal{V} \in \mathcal{R}^{6n}$ denoting the stacked vector of link spatial velocities, its spatial operator expression is [10]

$$\mathcal{V} = \phi^* H^* \dot{\theta} \tag{60}$$

Bundling together the rigid body transformations for all nodes we define the $\mathcal{B} \in \mathcal{R}^{6n \times 6n_c}$ *pick-off* matrix such that the stacked vector of node spatial velocities $\mathcal{V}_c$ can be expressed as

$$\mathcal{V}_c = \mathcal{B}^*\mathcal{V} \overset{60}{=} \mathcal{B}^*\phi^* H^*\dot{\theta} \quad \Rightarrow \quad \mathcal{J} \overset{47}{=} \mathcal{B}^*\phi^* H^* \tag{61}$$

This is the spatial operator expression for the $\mathcal{J}$ Jacobian matrix. Using this expression and Eq. 59 for the mass matrix inverse within Eq. 49 leads to the following expression for $\underline{\Lambda}$:

$$\underline{\Lambda} \overset{49}{=} \mathcal{J}\mathcal{M}^{-1}\mathcal{J}^* \overset{59}{=} \mathcal{B}^*\phi^* H^*(I - H\psi\mathcal{K})^*\mathcal{D}^{-1}(I - H\psi\mathcal{K})H\phi\mathcal{B} \tag{62}$$

Using the spatial operator identity [10, 21]

$$(\mathrm{I} - \mathrm{H}\psi\mathcal{K})\mathrm{H}\phi = \mathrm{H}\psi \tag{63}$$

in Eq. 62 leads to the following simpler expression for $\underline{\Lambda}$:

$$\underline{\Lambda} = \mathcal{B}^*\Omega\mathcal{B} \text{ with } \Omega \overset{\triangle}{=} \psi^*\mathrm{H}^*\mathcal{D}^{-1}\mathrm{H}\psi \in \mathcal{R}^{6n_c \times 6n_c} \tag{64}$$

We have arrived at an expression for $\underline{\Lambda}$, that unlike Eq. 49, involves neither the mass matrix inverse nor the node's Jacobian matrix! We refer to Ω as the *extended operational space compliance matrix*. This terminology is based on Eq. 64 which shows that the OSCM, $\underline{\Lambda}$ can be obtained by a reducing transformation of the full, all body Ω matrix by the $\mathcal{B}$ pick-off operator involving just the matrix sub-blocks associated with the parent links of the nodes. From its definition, it is clear that Ω is a symmetric and positive semi-definite since $\mathcal{D}^{-1}$ is a symmetric positive-definite matrix.

While the explicit computation of $\mathcal{M}^{-1}$ or $\mathcal{J}$ is not needed to obtain $\underline{\Lambda}$, the direct evaluation of Eq. 64 still remains of $O(\mathcal{N}^3)$ complexity due to the need for carrying out the multiple matrix/matrix products. The next section shows that these matrix/matrix products can be avoided by exploiting a decomposition of the Ω matrix.

10.3 Decomposition of Ω

The following lemma describes a decomposition of Ω into simpler component terms and an expression for its block elements. The $\mathcal{E}_\psi^*$ and $\psi()$ terms used below are defined in references [10, 20]. Furthermore, $\wp(k)$ denotes the parent link for the k^{th} link, and $i \prec j$ notation implies that the j^{th} link is an ancestor of the i^{th} link in the tree.

Lemma 1 **Decomposition of** Ω

Ω can be decomposed into the following disjoint sum:

$$\Omega = \Upsilon + \tilde{\psi}^*\Upsilon + \Upsilon\tilde{\psi} + R \quad \textit{where} \quad R \overset{\triangle}{=} \sum_{\substack{\forall i,j:\; i \nprec \nsucc j \\ k = \wp(i,j)}} e_i \psi^*(k,i) Y(k) \psi(k,j) e_j^* \tag{65}$$

$\Upsilon \in \mathcal{R}^{6n_c \times 6n_c}$ is a block-diagonal operator, referred to as the operational space compliance kernel, satisfying the following backward Lyapunov equation:

$$\mathrm{H}^*\mathcal{D}^{-1}\mathrm{H} = \Upsilon - \mathrm{diagOf}\left\{\mathcal{E}_\psi^*\Upsilon\mathcal{E}_\psi\right\} \tag{66}$$

diagOf $\left\{\mathcal{E}_\psi^*\Upsilon\mathcal{E}_\psi\right\}$ *represents just the block-diagonal part of the (generally non block-diagonal) $\mathcal{E}_\psi^*\Upsilon\mathcal{E}_\psi$ matrix. The* 6×6 *dimensional, symmetric, positive semi-definite $\Upsilon(k)$ diagonal matrices satisfy the following parent/child recursive relationship:*

$$\Upsilon(k) = \psi^*(\wp(k),k)\Upsilon(\wp(k))\psi(\wp(k),k) + \mathrm{H}^*(k)\mathcal{D}^{-1}(k)\mathrm{H}(k) \tag{67}$$

This relationship forms the basis for the following $O(\mathcal{N})$ base-to-tips scatter recursion for computing the $\Upsilon(k)$ diagonal elements:

$$\left\{\begin{array}{l} \textbf{for } \textit{all nodes } k \ \textit{(base-to-tips scatter)} \\ \quad \Upsilon(k) = \psi^*(\wp(k),k)\Upsilon(\wp(k))\psi(\wp(k),k) + \mathrm{H}^*(k)\mathcal{D}^{-1}(k)\mathrm{H}(k) \\ \textbf{end loop} \end{array}\right. \tag{68}$$

While Υ defines the block-diagonal elements of Ω, the following recursive expressions describe its off-diagonal terms:

$$\Omega(i,j) = \begin{cases} \Upsilon(i) & \textit{for } i = j \\ \Omega(i,k)\psi(k,j) & \textit{for } i \succeq k \succ j, \quad k = \wp(j) \\ \Omega^*(j,i) & \textit{for } i \prec j \\ \Omega(i,k)\psi(k,j) & \textit{for } i \nsucc j,\ j \nsucc i,\ k = \wp(i,j) \end{cases} \tag{69}$$

Proof: See [10, 20].

Eq. 65 shows that Ω can be decomposed into the sum of simpler terms consisting of the block diagonal Υ, the upper-triangular $\tilde{\psi}^*\Upsilon$, the lower triangular $\Upsilon\tilde{\psi}$, and the sparse R matrices. Furthermore, Eq. 69 reveals that all of the block-elements of $\Omega(i,j)$ can be obtained from the $\Upsilon(i)$ elements of the Υ block-diagonal operational space compliance kernel.

From the $\underline{\Lambda} = \mathcal{B}^*\Omega\mathcal{B}$ expression, and the sparse structure of $\mathcal{B}$, it is clear that only a subset of the elements of Ω are needed to compute $\underline{\Lambda}$. The $\mathcal{B}$ pick-off operator has one column for each of the nodes, with each such column having only a single non-zero 6×6 matrix entry at the k^{th} parent link slot. Only as many elements of Ω as there are elements in $\underline{\Lambda}$ are needed. Thus, just $n_c \times n_c$ number of 6×6 sub-block matrices of Ω are required. In view of the symmetry of the matrices, we actually need just $n_c(n_c + 1)/2$ such sub-block matrices. The overall complexity of this algorithm is linearly proportional to the number of degrees of freedom, and a quadratic function of the number of nodes. This is much lower than the $O(\mathcal{N}^3)$ complexity implied by Eq. 49.