

# Forward Dynamics Algorithms for Multibody Chains and Contact \*

Dinesh K. Pai, Uri M. Ascher, and Paul G. Kry  
Department of Computer Science  
University of British Columbia  
Vancouver, Canada  
{pai | ascher | pgkry}@cs.ubc.ca

## Abstract

*We describe a framework for derivation of several forward dynamics algorithms used in robotics. The framework is based on formulating an augmented system and performing block matrix elimination on this system. Several popular algorithms such as the  $O(N)$  Articulated Body method, and Composite Rigid Body method can be easily derived. We also derive an algorithm for simulation of contact between smooth bodies of arbitrary shape, in contact coordinates. Finally, we discuss some potential numerical difficulties that could arise and their solution.*

## 1 Introduction

During the last quarter century, robot dynamics algorithms have received considerable attention due to the central role they play in robot simulation, design, and control. See [2, 5, 6, 8, 10, 16, 21, 25, 26], and the references therein for related work. During this time, efficient algorithms were developed to compute the forward dynamics of robot arms — in  $O(N)$  time where  $N$  is the number of degrees of freedom [25, 2, 10]; we refer to them as the Articulated Body Methods (ABM). Alternatively, Composite Rigid Body Methods (CRBM) [26] are a family of  $O(N^3)$  methods which form the joint space inertia matrix explicitly [16]; these methods can be faster in practice than ABMs, for typical robot arms. Interest in robot dynamics algorithms has spread to other areas, most notably computer graphics and animation [7], and molecular dynamics [13].

The derivation of ABM algorithms, as the name implies, hinges on the concept of Articulated Body Inertias, whose connection to fast algorithms is subtle to non-experts in dynamics. An alternate approach us-

ing spatial operator algebras was proposed by [21, 12], who recognized the structural similarity of the ABM recursions to those arising in Kalman filtering. The operator algebra approach provides new insights into these algorithms and is more intuitive to those familiar with Kalman filtering.

In this paper we take an approach based on block matrix algorithms and elimination methods, which may be helpful to readers more comfortable with matrix computations. This approach was presented in [17], and subsequently used in [5] for analyzing the numerical stability of forward dynamics algorithms. The basic strategy is to formulate an augmented system of Differential Algebraic Equations (DAEs), which includes body variables, joint (constraint) variables, and constraint forces; then different dynamics algorithms can be derived in a straightforward manner, as different elimination methods for solving the resulting linear system. In particular, we show how the CRBM and ABM could be derived using this approach [5]. Articulated Body Inertias arise naturally as a side effect of the elimination process. In this paper we also show how the dynamic simulation of contact between two smooth objects of arbitrary shape fits in this framework, and derive an efficient algorithm for simulation in contact coordinates.

The rest of the paper is organized as follows. Section 2 introduces the notation we use throughout the paper. In Section 3 we describe the framework for deriving different forward dynamics algorithms. Section 4 derives a Composite Rigid Body method while Section 5 derives an Articulated Body method in this framework. In Section 6 we derive an algorithm for the dynamics of smooth rigid bodies in contact, using contact coordinates. Section 7 touches on the numerical difficulties that are faced in simulating certain ill-conditioned robots. Finally, we conclude with Section 8.

---

\*Supported in part by grants from IRIS (Institute for Robotics and Intelligent Systems) and NSERC.

## 2 Preliminaries

In the following we consider a chain of rigid bodies; we use a notation that is similar to [11, 22, 12, 18, 19].

The motion of a rigid body  $i = 1, \dots, n$  is described using a reference frame labeled  $i$  attached to the body. The homogeneous coordinates of the frame, relative to another frame  $j$ , is given by the  $4 \times 4$  matrix  ${}^j_i\mathbf{E}$ . We will always use leading subscripts and superscripts to indicate frames. The coordinates of a vector  $\mathbf{x}$ , in frame  $i$ , are given by the column matrix  ${}^i\mathbf{x}$ .

Sometimes it is necessary to define several frames associated with a single rigid body, for instance, when we treat contact problems; we label such frames with a trailing letter, e.g.,  $1c$ , is a frame “c” on body 1. We drop the frame label when it’s clear from context.

The spatial velocity [11]  $\phi(i, j)$  describes the relative motion of body  $j$  with respect to body  $i$ . In coordinates it’s given by the  $6 \times 1$  matrix  $\phi = (\omega^T, \mathbf{v}^T)^T$ , where  $\omega$  is the angular velocity and  $\mathbf{v}$  is the linear velocity of a point at the origin (see [11] for an interpretation of these quantities). Spatial wrenches are represented as  $\mathbf{f} = (f_r^T, f_t^T)^T$  where  $f_r$  is the torque and  $f_t$  is the (translational) force. Note that we do not reverse the order of forces and torques, unlike the practice in screw theory; we merely remember that these are covariant quantities which transform differently than contravariant quantities such as velocities. Spatial velocities transform according to the Adjoint transformation  ${}^j_i\text{Ad}$ :  ${}^j\phi = {}^j_i\text{Ad} {}^i\phi$ . If

$${}^j_i\mathbf{E} = \begin{pmatrix} \Theta & p \\ 0 & 1 \end{pmatrix},$$

then

$${}^j_i\text{Ad} = \begin{pmatrix} \Theta & 0 \\ [p]\Theta & \Theta \end{pmatrix}.$$

The bracket notation “[ $p$ ]” denotes the matrix of the cross product  $p \times$ . We define the spatial cross product of  $\phi = (\omega^T, \mathbf{v}^T)^T$  as the linear operator with coordinate matrix

$$[\phi] = \begin{pmatrix} [\omega] & 0 \\ [v] & [\omega] \end{pmatrix}.$$

## 3 A Framework for Forward Dynamics Algorithms

The Newton-Euler equations for rigid body  $i$  can now be written using spatial vectors as

$$f_i = M_i \dot{\phi}_i - [\phi_i]^T M_i \phi_i \quad (1)$$

where  $\phi_i$  is the spatial velocity of the body relative to the world (inertial) frame,  $M_i$  is the spatial inertia, and all coordinates are with respect to the body fixed frame.

Suppose the links of the robot are numbered, as in [22, 12, 5], starting from the free end (note that this is the opposite of the convention used elsewhere in the robotics literature). Further, we will assume that if the coordinate frame is not specified, a quantity for body  $k$  is expressed in frame  $k$  coordinates. Specifically, let  $\phi_k \stackrel{\text{def}}{=} {}^k\phi(0, k)$  be spatial velocity of  $k$ th body relative to the inertial frame (labeled 0),  $f_k \stackrel{\text{def}}{=} {}^k f_k$  be the wrench on body  $k$  due to a joint or contact constraint with body  $k+1$ , and  $M_k \stackrel{\text{def}}{=} {}^k M_k$  be the spatial inertia of  $k$ th body.

The equations of motion for the  $k$ th link are (compare Equation (1)),

$$f_k = {}_k^{k-1}\text{Ad}^T f_{k-1} + M_k \dot{\phi}_k + b_k \quad (2)$$

for  $k = 1, 2, \dots, n$ , ( $f_0 = 0$ ). Here  $b_k = -[\phi_k]^T M_k \phi_k$  is the gyroscopic force.

Constraints due to the joints are introduced as follows. Let  $H_k$  be the joint matrix for  $k$ th joint in frame  $k$  coordinates. For a 1 DOF joint,  $H_k$  is a  $6 \times 1$  matrix; if all joints have 1 DOF then  $N = n$ .

The relative spatial velocity across the  $k$ th joint is  $H_k \dot{q}_k$ ; therefore  $\phi_k = {}_k^{k+1}\text{Ad} \phi_{k+1} + H_k \dot{q}_k$ . Differentiating, and using D’Alembert’s principle,

$$\dot{\phi}_k = {}_k^{k+1}\text{Ad} \dot{\phi}_{k+1} + H_k \ddot{q}_k + a_k, \quad (3)$$

for  $k = n, n-1, \dots, 1$ , ( $\phi_{n+1} = \dot{\phi}_{n+1} = 0$ ). Here,  $a_k$  is Coriolis and centrifugal acceleration. We also write

$$\tau_k = H_k^T f_k, \quad (4)$$

where  $\tau_k$  is the joint force at  $k$ th joint (scalar for a 1 DOF joint).

We write equations (2), (3), and (4), involving  $\ddot{q}$ ,  $\dot{\phi}$  and  $f$  as one large algebraic system, following [17]. We illustrate the system matrices using 2 or 3 link chains so that the matrices fit within the columns of this document; the generalization to arbitrary kinematic chains is obvious. The system is now written as

$$M\mathbf{x} = \mathbf{b}, \quad (5)$$

where

$$M = \begin{pmatrix} I & M_1 & 0 & & & \\ 0 & I & H_1 & & -{}^1_2\text{Ad} & \\ H_1^T & 0 & 0 & & & \\ -{}^1_2\text{Ad}^T & & & I & M_2 & 0 \\ & & & 0 & I & H_2 \\ & & & H_2^T & 0 & 0 \end{pmatrix},$$

$$\mathbf{x} = \begin{pmatrix} f_1 \\ -\phi_1 \\ \ddot{q}_1 \\ f_2 \\ -\phi_2 \\ \ddot{q}_2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ -a_1 \\ \tau_1 \\ b_2 \\ -a_2 \\ \tau_2 \end{pmatrix}.$$

The matrix  $\mathbf{M}$  is a permutation of a symmetric matrix in KKT form (see [5]); here it has been arranged to make the connections to linear time dynamics algorithms clearer. Next, we consider different forward dynamics algorithms as strategies for solving this system.

## 4 The Composite Rigid Body Method

The Composite Rigid Body Method [26] (CRBM) is one approach to solving Equation (5). If we rearrange the system to group the joint accelerations together by column and row exchanges, we get

$$\left( \begin{array}{cc|cc|c} M_1 & & I & & \\ \hline & M_2 & -{}^1_2\text{Ad}^T & I & \\ \hline I & -{}^1_2\text{Ad} & & & H_1 \\ & I & & & H_2 \\ \hline & & H_1^T & & \\ & & & H_2^T & \end{array} \right) \quad (6)$$

The system can now be written as

$$\begin{pmatrix} M & A^{-T} & 0 \\ A^{-1} & 0 & H \\ 0 & H^T & 0 \end{pmatrix} \begin{pmatrix} -\dot{\phi} \\ f \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} b \\ -a \\ \tau \end{pmatrix} \quad (7)$$

where we have labeled the blocks delineated above as  $M$ ,  $A^{-1}$ ,  $H$ , etc.

Block-row elimination of this matrix produces

$$\begin{pmatrix} M & A^{-T} & 0 \\ A^{-1} & 0 & H \\ 0 & 0 & \mathcal{M} \end{pmatrix} \begin{pmatrix} -\dot{\phi} \\ f \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} b \\ -a \\ \tau - c \end{pmatrix}$$

where  $\mathcal{M} = H^T A^T M A H$  is the joint space inertia matrix and  $c = H^T A^T (M A a + b)$ . Hence, the last row can be solved for the joint accelerations  $\ddot{q}$ . This essentially gives the CRB method<sup>1</sup>. Since the  $N \times N$  matrix  $\mathcal{M}$  is dense, solving for the joint accelerations

<sup>1</sup>We note that the CRBM actually involves one more feature that is essential for the efficiency of the method: joint space inertia matrix  $\mathcal{M}$  is computed as  $\mathcal{M} = H^T (R + \tilde{A}^T R + R \tilde{A}) H$  where  $R$  is a block diagonal matrix of composite rigid body inertias and  $\tilde{A} = A - I$ .

takes  $O(N^3)$  time. Nevertheless, this method is the faster than ABM for small  $N$  because the constants involved in the complexity of the algorithm are small.

## 5 The Articulated Body Method

The Articulated Body Method [11] (ABM) can be derived from the same system matrix (5), but being careful to not lose the sparsity of the system, leading to an  $O(N)$  method. We have arranged Equation (5) to make this sparsity transparent and reveal the block diagonal structure. Straightforward block elimination with the smaller blocks in Equation (5) yields the ABM.

Each diagonal block corresponds to a body in the chain, which is coupled to the adjacent body via the matrices  ${}^k_{k+1}\text{Ad}^T$ . Elimination of these matrices, e.g., from the fourth block row in Equation (5), decouples adjacent bodies. As an algebraic side effect of this, the entry  $M_{k+1}$  is updated to

$$\hat{M}_{k+1} = M_{k+1} + {}^k_{k+1}\text{Ad}^T \hat{M}_k {}^k_{k+1}\text{Ad} - {}^k_{k+1}\text{Ad}^T \hat{M}_k H_k (H_k^T \hat{M}_k H_k)^{-1} H_k^T \hat{M}_k {}^k_{k+1}\text{Ad}$$

( $\hat{M}_1 = M_1$ ), and a corresponding update is applied to the right hand side. Thus,

$$\tau_k - c_k = \tau_k - H_k^T \hat{b}_k - H_k^T \hat{M}_k a_k$$

replaces  $\tau_k$  and

$$\hat{b}_{k+1} = b_{k+1} + {}^k_{k+1}\text{Ad}^T (\hat{b}_k + \hat{M}_k a_k) + {}^k_{k+1}\text{Ad}^T \hat{M}_k H_k (H_k^T \hat{M}_k H_k)^{-1} (\tau_k - c_k)$$

replaces  $b_{k+1}$ . We note that  $\hat{M}_k$  is precisely the articulated body inertia introduced by Featherstone [10]; the bias force  $z(k)$  in [12] is our  $\hat{b}_k + \hat{M}_k a_k$ .

For instance, in the  $n = 3$  case, we obtain the following structure of non-zero terms

$$\left( \begin{array}{cc|cc|c} I & M_1 & & & \\ & I & \times & & \\ & & D_1 & \times & \\ \hline & & & I & \hat{M}_2 \\ & & & I & \times \\ & & & & D_2 \\ \hline & & & & I & \hat{M}_3 \\ & & & & I & \times \\ & & & & & D_3 \end{array} \right) \quad (8)$$

Here,  $\times$  stands for a possibly nonzero block. The  $(3k, 3k)$  blocks are the joint inertias  $D_k = H_k^T \hat{M}_k H_k$  which are invertible. Solving this sparse triangular system yields, essentially, the Articulated Body method. The operation count is clearly  $O(N)$ . It is simple to extend this algorithm description to tree structured systems and even to systems with closed loops [17]. However, the latter lead to DAEs which cannot be simply reduced to ODEs, and the  $O(N)$  complexity is achieved only for a fixed number of closed loops, unless a special iterative method in time is used [4].

## 6 Contact Dynamics of Smooth Shapes

The dynamics of rigid bodies in contact also fits well in this framework; the additional complication is that the constraint matrices  $H$  are now more complex and depend on the contact configuration.

We allow the bodies to have smooth boundaries of *arbitrary* shape — for instance parametric surfaces (e.g., NURBS) or parametrizable shapes such as subdivision surfaces [23]. Montana [18] showed that if the surface is a differentiable manifold, the degrees of freedom of the contact can be explicitly parametrized, and derived the kinematic relationship between these *contact coordinates* and the motion of the contacting bodies. The contact can be considered a generalized joint [20, 16]. The contact equations have been used by [14] for estimating pose and motion from contact measurements and by [15] for motion planning. See also [1, 24].

However, only simple surfaces such as spheres and cylinders were used in previous work. One difficulty is the complexity of formulating the dynamics with more complex surfaces — treating the system as independent rigid bodies with explicit contact constraints leads to a high index DAE system. Solution methods then involve constraint differentiation and stabilization [3]. An attractive alternative is to formulate the dynamics in contact coordinates, which has similar benefits to using joint coordinates for robot arms. This fits in the current framework, which can be used to formulate the equations of motion in contact coordinates.

We first formulate the contact coordinates following [18, 19, 14]. For concreteness, we consider the case of two rigid bodies, with reference frames  $1$  and  $2$  (see Figure 1).

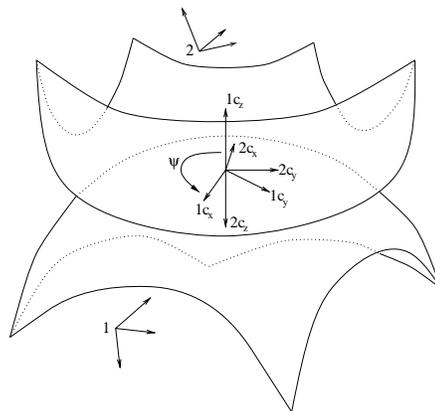


Figure 1: Contact Coordinates

Dealing with a chain of rigid bodies in contact is straightforward and completely analogous to the treatment in Section 5. A direct extension to tree-structured systems is also possible. Multiple contacts between a pair of bodies and closed chains can be handled as DAE's but can't be easily reduced to ODEs (e.g., [24]); we do not treat that case here.

Suppose the two bodies make contact at a single point with coordinates  ${}^1\mathbf{c}$  in frame  $1$  and  ${}^2\mathbf{c}$  in frame  $2$ . We assume that each surface is equipped with an atlas of coordinate charts; in the case of parametric surfaces, these charts are used to define the surface. Consider the surface of body  $1$ . Let  $u_1 = (u_1^s, u_1^t)^T \in \mathbb{R}^2$  be the coordinates of  ${}^1\mathbf{c}$  in a suitable chart, i.e.,  ${}^1\mathbf{c} = c(u_1)$ . We can define a normalized orthogonal frame, labeled  $1c$ , as follows. We use the notation  $c_{,s} \stackrel{\text{def}}{=} \frac{\partial c}{\partial s}$ . Let

$$\begin{aligned} {}^1\mathbf{x} &= \frac{c_{,s}}{\|c_{,s}\|}, \\ {}^1\mathbf{y} &= \frac{(c_{,s} \cdot c_{,s})c_{,t} - (c_{,s} \cdot c_{,t})c_{,s}}{\|(c_{,s} \cdot c_{,s})c_{,t} - (c_{,s} \cdot c_{,t})c_{,s}\|}, \\ {}^1\mathbf{z} &= \frac{c_{,s} \times c_{,t}}{\|c_{,s} \times c_{,t}\|}. \end{aligned}$$

The homogeneous coordinates of the frame  $1c$  with respect to frame  $1$  are given by

$${}^1_{1c}\mathbf{E} = \begin{pmatrix} {}^1\mathbf{x} & {}^1\mathbf{y} & {}^1\mathbf{z} & {}^1\mathbf{c} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

We can similarly assign an *orthonormal* frame on body  $2$  at contact point  ${}^2\mathbf{c}$ . For contact between smooth surfaces, the contact frames can be aligned by a rotation

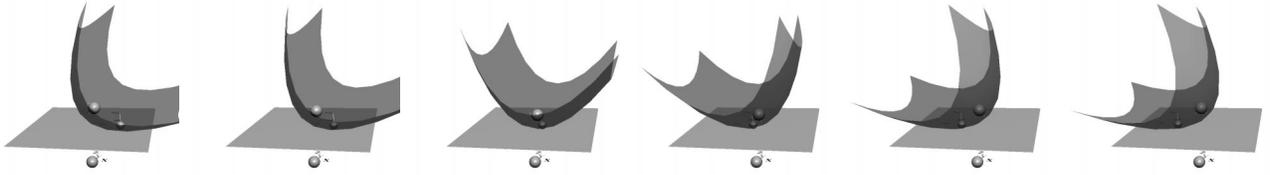


Figure 2: Simulation of frictionless contact with curved cubic Beziér surfaces. A large sphere and a coordinate frame are drawn at the center of mass of each object. The contact between the two surfaces is marked with a smaller sphere. Transparency is used to allow the contact position to be seen at all times. The initial spatial velocity of the curved patch is zero, thus the energy of the system is determined by the initial height of the center of mass of the upper patch. As the system evolves, the upper patch slips on the lower patch. As the potential energy converts to kinetic energy, the object rotates and translates, but the center of mass correctly moves only in the vertical direction.

matrix (see Figure 1), given by

$${}_{2c}^1\mathbf{E} = \begin{pmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ -\sin \psi & -\cos \psi & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} R_\psi & 0 \\ 0 & 1 \end{pmatrix}$$

Note that  ${}_{2c}^1\mathbf{E}$  is idempotent. The contact coordinates,  $q$ , are the patch coordinates of the contact point on the two bodies, and the rotation angle  $\psi$ ; i.e.,  $q = (u_1^T, u_2^T, \psi)^T$ . They play the same role as robot joint angles in forward dynamics computations.

As the contact point moves, so do the orthonormal frames at the contact points. The spatial velocity of the contact frame  $1c$  relative to the body frame  $1$  can be computed by differentiation (see, e.g., [18], for the general procedure), and has a particularly simple form in the contact frame

$${}_{1c}^1\phi(1, 1c) \stackrel{\text{def}}{=} {}^{1c}H_1 \dot{u}_1 \quad (9)$$

The matrix  $H_1$  has size  $6 \times 2$ . We can analogously define  ${}^{2c}H_2$ . This transforms to frame  $1c$  as

$${}^{1c}H_2 = {}^{1c}\text{Ad} \, {}^{2c}H_2 = \begin{pmatrix} R_\psi & 0 \\ 0 & R_\psi \end{pmatrix} {}^{2c}H_2. \quad (10)$$

Finally, relative spatial velocity of the two contact frames is a pure rotation about the surface normal, i.e.,

$${}^{1c}\phi(1c, 2c) = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \dot{\psi} \stackrel{\text{def}}{=} {}^{1c}H_\psi \dot{\psi} \quad (11)$$

We can now compute the contact matrices  $H_k$  as follows. The relative spatial velocity of the two bodies is

given by

$$\begin{aligned} \phi(1, 2) &= \phi(1, 1c) + \phi(1c, 2c) + \phi(2c, 2) \\ &= \phi(1, 1c) + \phi(1c, 2c) - \phi(2, 2c) \end{aligned}$$

Substituting Equations (9)–(11) we have

$${}^{1c}\phi(1, 2) = ({}^{1c}H_1 \quad -{}^{1c}H_2 \quad {}^{1c}H_\psi) \dot{q} \stackrel{\text{def}}{=} {}^{1c}H \dot{q} \quad (12)$$

The contact matrix  $H$  can now be transformed to any convenient frame, for instance, frame  $1$ :  ${}^1H = {}_{1c}^1\text{Ad} \, {}^{1c}H$ .

Now, as in Equation (5), we can write the equations of motion and the contact constraints as follows (for frictionless contact). For example, let body  $1$  be free to move in contact with body  $2$  which is fixed. All quantities below are expressed in frame  $1$  coordinates, so we drop the frame label for clarity.

$$\begin{pmatrix} I & M & 0 \\ 0 & I & H \\ H^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{f} \\ -\dot{\phi} \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} b \\ -a \\ 0 \end{pmatrix}. \quad (13)$$

Performing block row elimination on the last row produces

$$\begin{pmatrix} I & M & 0 \\ 0 & I & H \\ 0 & 0 & H^T M H \end{pmatrix} \begin{pmatrix} \dot{f} \\ -\dot{\phi} \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} b \\ -a \\ r \end{pmatrix}. \quad (14)$$

Here,

$$r = H^T (f_{ext} + [\phi]^T M \phi - M \dot{H} \dot{q})$$

and  $f_{ext}$  is the external force acting on the body.

The last row yields an ODE in the contact coordinates which can be integrated directly. Note that because contact coordinates are used, the constraints are automatically satisfied and there is no need for constraint

stabilization. Backsubstituting  $\ddot{q}$  into the second row, we get the contact wrench  $f$  which can be monitored to detect contact breaking.

We have implemented this algorithm in a dynamics simulator written in Java and using Java3D for animation. We use an explicit fourth order Runge-Kutta method to integrate the contact coordinate accelerations and velocities.

Figure 2 shows several frames from a simulation involving a curved patch on a fixed frictionless flat surface. The surfaces in this simulation are cubic Beziér tensor product patches. Running the HotSpot Java virtual machine on a dual Pentium II 450Mhz machine this computation takes about 1.2 ms without code optimizations.

## 7 Numerical Issues

The matrix factorization approach used in this paper has another advantage: it makes numerical difficulties with forward dynamics algorithms more explicit and makes it easier to address these difficulties. In [5] we analyzed a phenomenon called ‘‘Formulation Stiffness’’ in [9]. For some robot arms, dynamics algorithms such as the Composite Rigid Body method can produce severe cancellation errors when computed with finite precision arithmetic; the Articulated Body Method is more numerically well behaved, but can also exhibit similar difficulties. These numerical errors can slow down popular adaptive step-size integrators, or even make them fail.

In [5], we analyzed the problem as caused by different ways in which the pivot  $D_k = H_k^T \hat{M}_k H_k$  in Equation (8) is computed. This term also appears in the composite rigid body method when the joint space inertia  $\mathcal{M}$  is factored.

Suppose we have a body with a small mass matrix  $M_2$  connected to a body with large mass matrix  $M_1$ . The problem with the CRBM is that while forming  $\mathcal{M}$ , the small  $M_2$  is added to the large  ${}^1_2\text{Ad}^T M_1 {}^1_2\text{Ad}$ , and then a large term  ${}^1_2\text{Ad}^T M_1 H_1 (H_1^T M_1 H_1)^{-1} H_1^T M_1 {}^1_2\text{Ad}$  is subtracted during the solution process. The ABM typically does better, by first forming the projected inertia matrix  $\tilde{M}_2 = {}^1_2\text{Ad}^T (M_1 - M_1 H_1 (H_1^T M_1 H_1)^{-1} H_1^T M_1) {}^1_2\text{Ad}$ , and then adding it to  $M_2$ . This rank deficient matrix  $\tilde{M}_2$  can have large terms, and yet the contribution of  $M_2$  to its null space is significant.

In fact, we can go a step further; it is possible that the rank deficient term  $\tilde{M}_2$  has all large entries because

the coordinate axes are not aligned with its null space. To compute the pivot  $D_2 = H_2^T \hat{M}_2 H_2$  more stably, we can form it as

$$D_2 = H_2^T M_2 H_2 + H_2^T \tilde{M}_2 H_2.$$

This has the additional cost of computing a quadratic form with  $H_2$  twice, but it further reduces cancellation errors because it preserves the contribution of  $M_2$  to the projection of  $\hat{M}_2$  onto the  $H_2$  joint; indeed this is where it matters. The resulting algorithm is the Modified ABM proposed in [5].

## 8 Conclusions

We have shown how several, apparently different, robot dynamics algorithms can be viewed as different ways of solving the same matrix equation. This approach is straightforward to apply, and reveals the connection between, for instance, the Articulated Body dynamics algorithm and our algorithm for simulating the dynamics of contact between arbitrary smooth surfaces. We also discuss some numerical difficulties that could arise in forward dynamics algorithms and a modified ABM algorithm to address these difficulties.

## References

- [1] M. Anitescu, J. Cremer, and F. Potra. Formulating 3d contact dynamics problems. *Mechanics of Structures and Machines*, 24(4):405–437, 1996.
- [2] W.W. Armstrong. Recursive solution to the equations of motion of an n-link manipulator. In *Proceedings of the 5th World Congress on Theory of Machines and Mechanisms*, volume 2, pages 1343–1346, Montreal, 1979.
- [3] U. Ascher. Stabilization of invariants of discretized differential systems. *Numerical Algorithms*, 14:1–23, 1997.
- [4] U. Ascher and P. Lin. Sequential regularization methods for simulating mechanical systems with many closed loops. *SIAM J. Scient. Comput.*, 21:1244–1262, 1999.
- [5] U. Ascher, D. K. Pai, and B. Cloutier. Forward dynamics, elimination methods, and formulation stiffness in robot simulation. *International Journal of Robotics Research*, 16(6):749–758, December 1997.
- [6] D.S. Bae and E.J. Haug. A recursive formulation for constrained mechanical system dynamics: Part i. open loop systems. *Mechanical Structures & Machines*, 15(3):359–382, 1987.

- [7] David Baraff. Linear-Time simulation using lagrange multipliers. In *SIGGRAPH 96 Conference Proceedings*, pages 137–146, 1996.
- [8] H. Brandl, R. Johanni, and M. Otter. A very efficient algorithm for the simulation of robots and similar multibody systems without inversion of the mass matrix. In *Proceedings of IFAC/IFIP/IMACS International Symposium on the Theory of Robots*, Vienna, Austria, December 1986.
- [9] B. Cloutier, D. K. Pai, and U. M. Ascher. The formulation stiffness of forward dynamics algorithms and implications for robot simulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2816–2822, May 1995.
- [10] R. Featherstone. The calculation of robot dynamics using articulated-body inertias. *International Journal of Robotics Research*, 2(1):13–30, Spring 1983.
- [11] Roy Featherstone. *Robot dynamics algorithms*. Kluwer, 1987.
- [12] A. Jain. Unified formulation of dynamics for serial rigid multibody systems. *Journal of Guidance, Control, and Dynamics*, 14(3):531–542, 1991.
- [13] A. Jain, N. Vaidehi, and G. Rodriguez. A fast recursive algorithm for molecular dynamics simulations. *Journal of Computational Physics*, 23:239–248, June 1993.
- [14] Yan-Bin Jia and Michael A. Erdmann. Observing pose and motion through contact. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1998.
- [15] Z. Li and J. F. Canny. Motion of two rigid bodies with rolling constraint. *IEEE Transactions on Robotics and Automation*, 6(1):62–72, 1990.
- [16] Kathryn W. Lilly. *Efficient dynamic simulation of robotic mechanisms*. Kluwer, 1993.
- [17] C. Lubich, U. Nowak, U. Pohle, and Ch. Engstler. Mexx – numerical software for the integration of constrained mechanical multibody systems. Tech. report SC92-12, Konrad-Zuse-Zentrum für Informationstechnik, Berlin., 1992.
- [18] David J. Montana. The kinematics of contact and grasp. *International Journal of Robotics Research*, 7(3):17–32, 1988.
- [19] R. Murray, Z. Li, and S. S. Sastry. *A mathematical introduction to robotic manipulation*. CRC Press, 1994.
- [20] R. E. Roberson and R. Schwertassek. *Dynamics of Multibody Systems*. Springer-Verlag, 1988.
- [21] G Rodriguez. Kalman filtering, smoothing and recursive robot arm forward and inverse dynamics. *IEEE Journal of Robotics and Automation*, 3(6):624–639, 1987.
- [22] G. Rodriguez, A. Jain, and K. Kreutz-Delgado. A spatial operator algebra for manipulator modeling and control. *The International Journal of Robotics Research*, 10(4):371–381, 1991.
- [23] Jos Stam. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *SIGGRAPH 98 Conference Proceedings*, pages 395–404, 1998.
- [24] J. C. Trinkle, J. S. Pang, S. Sudarsky, and G. Lo. On dynamic multi-rigid-body contact problems with coulomb friction. To appear in *Zeitschrift für Angewandte Mathematik und Mechanik.*, 1999.
- [25] A.F. Vereshchagin. Computer simulation of the dynamics of complicated mechanisms of robot-manipulators. *Engineering Cybernetics*, 6:65–70, 1974.
- [26] M. W. Walker and D. E. Orin. Efficient dynamic computer simulation of robotic mechanism. *ASME J. Dynamic Systems, Measurement and Control*, 104:205–211, 1982.