# A Technique for Analysing Constrained Rigid-Body Systems, and its Application to the Constraint Force Algorithm

Roy Featherstone*and Amir Fijany†

December 2, 1999

**Abstract**

*Roy Featherstone is currently with the Department of Computer Science, University of Wales Aberystwyth, Penglais, Aberystwyth SY23 3DB, Wales, UK. At the time this work was done, he was with the Department of Engineering Science, Oxford University,
and was supported by EPSRC Advanced Research Fellowship number B92/AF/1466.

†The research of Amir Fijany was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

The Constraint Force Algorithm, as originally described by Fijany et al., calculates the forward dynamics of a system comprising $N$ rigid bodies connected together in an unbranched chain with joints from a restricted class of joint types. It was designed for parallel calculation of the dynamics, and achieves $O(\log N)$ time complexity on $O(N)$ processors. This paper presents a new formulation of the Constraint Force Algorithm that corrects a major limitation in the original, and sheds new light on the relationship between it and other dynamics algorithms. The new version is applicable to systems with any type of joint, floating bases, and short branches off the main chain. It is obtained using a new technique for analysing constrained rigid-body systems by means of a change of basis in a dual system of vector spaces. This new technique is also described.

# 1 Introduction

The Constraint Force Algorithm (CFA) was the first algorithm to calculate the forward dynamics of an $N$-body robot manipulator in $O(\log N)$ time on a parallel computer with $O(N)$ processors. The original version, as described in [1], was applicable to a system comprising a fixed base and $N$ rigid bodies, connected together in an unbranched chain by joints from a restricted class of joint types. This was subsequently extended to floating bases in [2].

This paper presents a new formulation of the CFA that corrects a major limitation in the original formulation, and sheds new light on the relationship between the CFA and other dynamics algorithms. It also presents an improved method for dealing with floating bases that is easier and more efficient than the method described in [2]; and it extends the CFA to branched kinematic trees consisting of a single main chain and any number of short side-branches. Floating bases are implemented by means of a 6-DoF joint, and short branches are implemented using articulated-body techniques.

The original formulation, as described in [1], includes an incorrect usage of orthogonal complements. Specifically, the inner product that is used to define orthogonality is non-invariant and dimensionally inconsistent. See [3] for a full explanation of the problem. In [1], the problem is finessed by observing that if the algorithm is restricted to certain types of joint then it is possible to formulate the affected equations (Eqs. 6 and 7) in such a way that the coefficients of the dimensionally-inconsistent terms are zero. The new formulation removes the source of the problem by avoiding orthogonal complements altogether. The immediate result is to remove all restrictions on joint type.

The new formulation uses a 'change of basis' (CoB) technique that has not been previously published, so a brief description is included in this paper. The method can be summarized as follows: given the equation of motion of a rigid-body system in the form of a linear equation between acceleration and force, plus an acceleration constraint to be imposed on the system, it is possible to construct new bases on the force and acceleration vector spaces such that the equation of motion is decoupled into two independent subsystems, one of which is completely immobilized by the given constraint, while the other is completely unconstrained. Because the two subsystems are independent, a constraint imposed on one has no effect on the other.

This technique can be used both to derive the CFA and to derive conventional dynamics algorithms; and it reveals a simple relationship between the two via an equation that can be paraphrased as 'constrained dynamics = unconstrained dynamics − immobilized dynamics'. Conventional algorithms are derived from the LHS of this equation, and the CFA from the RHS.

The CoB technique does, of course, build on much existing work. The idea of using separate vector spaces for motion and force vectors originates in analytical mechanics; and the idea of using separate motion and force spaces to describe the freedom and constraint spaces of a robot in contact with its environment has become widespread in the area of hybrid motion/force control [4, 5, 6]. Earlier works in this area tended to use a single space containing both types of vector, and some of them suffer from an incorrect use of orthogonal complements; but a correct version of the single-space analysis can be found in [7, 8]. The argument in favour of two spaces is put in [5]. These works partly inspired the present work.

The method of constructing subspaces of freedom and constraint is also not new. Although the details differ in each case, the same basic method is apparent in [9, 7, 8, 4, 5, 6]. The method presented here is slightly more general in that we do not require the parent spaces to have the special properties of twists and wrenches, do not require the dimension to be 6, and place no limit on the number of bodies that are subject to constraints.

On the other hand, the particular decoupling presented in this paper is quite different from that presented in [6], which is done using eigenvalues and eigenvectors, and solves a different problem.

The rest of this paper is organized into three sections. The first describes the CoB technique in general terms; the second shows how it is applied to constrained rigid-body dynamics; and the third describes the new CFA formulation.

## 2 The Change-of-Basis Technique

Let $U$ and $V$ be two $n$-dimensional vector spaces with a scalar product defined between them. This product takes one argument from each space; so, if $\mathbf{u} \in U$ and $\mathbf{v} \in V$, then $\mathbf{u}\cdot\mathbf{v}$ is defined, but $\mathbf{u}\cdot\mathbf{u}$ and $\mathbf{v}\cdot\mathbf{v}$ are not. The expression $\mathbf{v}\cdot\mathbf{u}$ is permitted as a synonym for $\mathbf{u} \cdot \mathbf{v}$. A mathematical structure comprising these two spaces and their scalar product is called a system of dual vector spaces, and may be denoted $\langle U, V, \cdot\rangle$.

If $\mathbf{u} \cdot \mathbf{v} = 0$ then $\mathbf{u}$ and $\mathbf{v}$ are said to be reciprocal. They cannot be said to be orthogonal, because orthogonality is a relationship defined between members of the same space. The term 'reciprocal' has been borrowed from screw theory in order to avoid the term 'orthogonal'; but note that its meaning here differs from its usual meaning in screw theory, where it has a geometrical interpretation based on the special properties of screws, and where concepts like self-reciprocity are defined.

Two subspaces, $S \subseteq U$ and $T \subseteq V$, are reciprocal if every element of $S$ is reciprocal to every element of $T$, and this relationship is denoted $S \perp T$. In general, infinitely many spaces $T$ satisfy $S \perp T$ for a given $S$; but the space of largest dimension is unique, and is called the reciprocal complement of $S$, denoted $S^{\perp}$. An alternative definition is

$$S^{\perp} = \{\mathbf{v} \mid \forall \mathbf{u} \in S,\ \mathbf{u} \cdot \mathbf{v} = 0\} \subset V.$$

The sum of the dimensions of $S$ and $S^{\perp}$ is $n$.

Let us now introduce some bases on $U$ and $V$. Let $\{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ be an arbitrary basis on $U$. For each such basis, there exists a unique basis $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ on $V$ with the property

$$\mathbf{d}_i \cdot \mathbf{e}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}.$$

A pair of bases that satisfy this condition is called a reciprocal basis-pair. If $[\mathbf{u}]$ and $[\mathbf{v}]$ are $n \times 1$ matrices of coordinates representing $\mathbf{u}$ and $\mathbf{v}$ in bases that form a reciprocal pair then $\mathbf{u} \cdot \mathbf{v} = [\mathbf{u}]^T [\mathbf{v}]$.

The reciprocal basis-pair plays a role similar to that of an orthonormal basis in an inner product space, but there are important differences. In particular, there are $n^2$ freedoms available in choosing a reciprocal basis-pair in $\langle U, V, \cdot \rangle$, but only $n(n-1)/2$ freedoms available in choosing an orthonormal basis on an $n$-dimensional inner product space. These extra freedoms are

-bb-error = =

Figure 1: Orthonormal basis (a) vs. reciprocal basis-pair (b).

essential to the success of the CoB technique.

The difference is illustrated by the 2-D example shown in Figure 1. The orthonormal basis (a) consists of two unit vectors at right angles. As there is no freedom to alter the lengths of the vectors or the angle between them, the only remaining freedom is the overall orientation of the basis. On the other hand, the reciprocal basis-pair (b) consists of any two linearly-independent vectors in one space (*e.g.* $\mathbf{d}_1$ and $\mathbf{d}_2$) plus two vectors in the other space that are uniquely determined by the reciprocity conditions. In this case, there are four freedoms available. Although the concepts of magnitude and angle are not, in general, defined on $\langle U, V, \cdot \rangle$, we have used them as visual cues to illustrate the reciprocity conditions: $\mathbf{d}_1$ is shown at right angles to $\mathbf{e}_2$ to indicate $\mathbf{d}_1 \cdot \mathbf{e}_2 = 0$, and so on.

From here on, we switch from using abstract vectors (also known as coordinate-free vectors) to using coordinate-based representations of vectors, and assume that a reciprocal basis-pair is being used. To avoid a messy change of notation (*e.g.* from $\mathbf{u}$ to $[\mathbf{u}]$), all symbols denoting vectors will now refer to coordinate vectors, except where explicitly stated to the contrary.

Once we have coordinate vectors, we can define matrices to represent subspaces. If $S$ is an $r$-dimensional subspace of $U$ then it can be represented by any $n \times r$ matrix $\mathbf{S}$ satisfying $S = \text{Range}(\mathbf{S})$. If $\mathbf{S}_0$ is one such matrix then all others can be expressed in the form $\mathbf{S} = \mathbf{S}_0\,\mathbf{A}$ where $\mathbf{A}$ is any nonsingular $r \times r$ matrix. The columns of $\mathbf{S}$ can be thought of as a set of basis vectors on $S$; and any element of $S$ can be expressed in the form $\mathbf{S}\,\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is an $r$-dimensional vector of coordinates. If two subspaces $S$ and $T$ satisfy $S \perp T$ then any two matrices representing them satisfy $\mathbf{S}^T\,\mathbf{T} = \mathbf{0}$.

Let $B_1$ and $B_2$ be two reciprocal basis-pairs in $\langle U, V, \cdot \rangle$, and let $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{v}_1$ and $\mathbf{v}_2$ be representations of the abstract vectors $\mathbf{u}$ and $\mathbf{v}$ in $B_1$ and $B_2$. The transformation rules for changes of basis are

$$\mathbf{u}_2 = \mathbf{X}_U\,\mathbf{u}_1, \quad \mathbf{u}_1 = \mathbf{X}_U^{-1}\,\mathbf{u}_2,$$

$$\mathbf{v}_2 = \mathbf{X}_V\,\mathbf{v}_1, \quad \mathbf{v}_1 = \mathbf{X}_V^{-1}\,\mathbf{v}_2,$$

where $\mathbf{X}_U$ is a coordinate transformation matrix that performs the change of basis in $U$, and $\mathbf{X}_V$ does the same in $V$. The abstract scalar product, $\mathbf{u} \cdot \mathbf{v}$, is inherently invariant with respect to changes of basis; so we know that $\mathbf{u}_1^T \mathbf{v}_1 = \mathbf{u}_2^T \mathbf{v}_2$ for all $B_1$ and $B_2$, hence

$$\mathbf{X}_U^T \equiv \mathbf{X}_V^{-1}.$$

With two vector spaces, there are four types of linear mapping that we can define: $U \mapsto U$, $U \mapsto V$, $V \mapsto U$ and $V \mapsto V$. Each has its own transformation rule, which is easily deduced from the corresponding rules for vectors:

$$U \mapsto U: \quad \mathbf{A}_2 = \mathbf{X}_U \, \mathbf{A}_1 \, \mathbf{X}_U^{-1}$$

$$U \mapsto V: \quad \mathbf{B}_2 = \mathbf{X}_V \, \mathbf{B}_1 \, \mathbf{X}_U^{-1}$$

$$V \mapsto U: \quad \mathbf{C}_2 = \mathbf{X}_U \, \mathbf{C}_1 \, \mathbf{X}_V^{-1}$$

$$V \mapsto V: \quad \mathbf{D}_2 = \mathbf{X}_V \, \mathbf{D}_1 \, \mathbf{X}_V^{-1}.$$

Let us now move on to the CoB technique itself. Suppose we are given a symmetric, positive-definite mapping $\mathbf{M} : U \mapsto V$ and an arbitrary subspace $S_1 \subset U$. It is always possible to find three additional subspaces, $S_2$, $T_1$ and $T_2$, satisfying the following equations:

$$\begin{aligned} S_1 \oplus S_2 = U, &\quad T_1 \oplus T_2 = V, \\ T_1 = \mathbf{M}\, S_1, &\quad T_2 = \mathbf{M}\, S_2, \\ S_1 \perp T_2, &\quad T_1 \perp S_2, \end{aligned} \tag{1}$$

where $\oplus$ means direct sum and $\mathbf{M}\, S_1$ is the image of $S_1$ under the mapping $\mathbf{M}$. The three subspaces are uniquely determined by the problem, but not their matrix representations. In fact, there are enough additional freedoms in the matrix representations to allow us to impose the following additional constraint:

$$[\mathbf{S}_1\, \mathbf{S}_2]^T\, [\mathbf{T}_1\, \mathbf{T}_2] = \mathbf{1}_{n\times n}.$$

With this extra condition, the solution is

$$\mathbf{T}_2 = \mathbf{S}_1^{\perp},$$

$$\mathbf{T}_1 = \mathbf{M}\,\mathbf{S}_1\,(\mathbf{S}_1^T\mathbf{M}\,\mathbf{S}_1)^{-1}, \tag{2}$$

$$\mathbf{S}_2 = \mathbf{M}^{-1}\,\mathbf{T}_2\,(\mathbf{T}_2^T\mathbf{M}^{-1}\mathbf{T}_2)^{-1}.$$

There is still a degree of arbitrariness in calculating $\mathbf{T}_2$ from $\mathbf{S}_1$, but $\mathbf{T}_1$ is now uniquely determined, and $\mathbf{S}_2$ is unique for a given $\mathbf{T}_2$. Incidentally, it is not necessary to start with $S_1$; we could have started with any one of the other three spaces.

In solving this problem, we have in effect defined a new reciprocal basis-pair with basis vectors $\mathbf{s}_i$ in $U$ and $\mathbf{t}_i$ in $V$, where $\mathbf{s}_i$ and $\mathbf{t}_i$ are the $i^{th}$ columns of the $n \times n$ matrices $[\mathbf{S}_1\ \mathbf{S}_2]$ and $[\mathbf{T}_1\ \mathbf{T}_2]$ respectively. This basis-pair has two special properties:

1. the first $r$ basis vectors in $U$ span the given subspace $S_1$, and
2. the given linear mapping is block-diagonal, comprising an $r \times r$ block that maps $S_1$ to $T_1$ and an $(n-r) \times (n-r)$ block that maps $S_2$ to $T_2$.

The first property is obvious. To see the second, simply transform $\mathbf{M}$ to the special basis-pair. The coordinate transformation matrices are

$$\mathbf{X}_U = [\mathbf{T}_1\, \mathbf{T}_2]^T, \quad \mathbf{X}_V = [\mathbf{S}_1\, \mathbf{S}_2]^T,$$

$$\mathbf{X}_U^{-1} = [\mathbf{S}_1\, \mathbf{S}_2], \quad \mathbf{X}_V^{-1} = [\mathbf{T}_1\, \mathbf{T}_2],$$

and the value of $\mathbf{M}$ in the special basis-pair is (using Eq. 2)

$$\mathbf{X}_V\, \mathbf{M}\, \mathbf{X}_U^{-1} = \begin{bmatrix} \mathbf{S}_1^T \mathbf{M}\, \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & (\mathbf{T}_2^T \mathbf{M}^{-1} \mathbf{T}_2)^{-1} \end{bmatrix}.$$

If we separate the two blocks and transform the whole equation back to the original basis-pair, the result is

$$\begin{aligned} \mathbf{M} &= \mathbf{X}_V^{-1} \left( \begin{bmatrix} \mathbf{S}_1^T \mathbf{M}\, \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\mathbf{T}_2^T \mathbf{M}^{-1} \mathbf{T}_2)^{-1} \end{bmatrix} \right) \mathbf{X}_U \\ &= \mathbf{M}\, \mathbf{S}_1\, (\mathbf{S}_1^T \mathbf{M}\, \mathbf{S}_1)^{-1}\, \mathbf{S}_1^T\, \mathbf{M} + \mathbf{T}_2\, (\mathbf{T}_2^T \mathbf{M}^{-1} \mathbf{T}_2)^{-1}\, \mathbf{T}_2^T. \end{aligned}$$

This equation expresses $\mathbf{M}$ as the sum of two components: one that maps $S_1$ to $T_1$ and one that maps $S_2$ to $T_2$. The equivalent expression for $\mathbf{M}^{-1}$ is obtained by pre- and post-multiplying this equation by $\mathbf{M}^{-1}$:

$$\mathbf{M}^{-1} = \mathbf{S}_1\,(\mathbf{S}_1^T\mathbf{M}\,\mathbf{S}_1)^{-1}\,\mathbf{S}_1^T + \mathbf{M}^{-1}\,\mathbf{T}_2\,(\mathbf{T}_2^T\mathbf{M}^{-1}\mathbf{T}_2)^{-1}\,\mathbf{T}_2^T\,\mathbf{M}^{-1}. \tag{3}$$

# 3 Applying CoB to Constrained Dynamics

Now let us apply the CoB technique to the dynamics of a physical system. Suppose we are given an unconstrained system for which the equation of motion is $\mathbf{u} = \mathbf{M}^{-1}\,\mathbf{v}$, where $\mathbf{u} \in U$ is the (unknown) output variable and $\mathbf{v} \in V$ the (known) input variable; and suppose this system is subjected to a known constraint in the form $\mathbf{u} \in S_1 \subset U$. The equation of motion of the constrained system is

$$\mathbf{u} = \mathbf{M}^{-1}\,(\mathbf{v} + \mathbf{v}_c),$$

where $\mathbf{v}_c \in V$ is the unknown, constraint-maintaining input, which is assumed to obey the principle of virtual work. Transforming this equation to the special basis-pair produces

$$\begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \end{bmatrix} = \begin{bmatrix} (\mathbf{S}_1^T\mathbf{M}\,\mathbf{S}_1)^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2^T\mathbf{M}^{-1}\mathbf{T}_2 \end{bmatrix} \left( \begin{bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\gamma}_1 \\ \boldsymbol{\gamma}_2 \end{bmatrix} \right),$$

where

$$\begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \end{bmatrix} = \mathbf{X}_U\,\mathbf{u}, \quad \begin{bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{bmatrix} = \mathbf{X}_V\,\mathbf{v}, \quad \begin{bmatrix} \boldsymbol{\gamma}_1 \\ \boldsymbol{\gamma}_2 \end{bmatrix} = \mathbf{X}_V\,\mathbf{v}_c.$$

Because the system matrix is block-diagonal, this equation actually represents two independent subsystems:

$$\boldsymbol{\alpha}_1 = (\mathbf{S}_1^T \mathbf{M}\,\mathbf{S}_1)^{-1}\,(\boldsymbol{\beta}_1 + \boldsymbol{\gamma}_1), \tag{4}$$

$$\boldsymbol{\alpha}_2 = \mathbf{T}_2^T \mathbf{M}^{-1} \mathbf{T}_2\,(\boldsymbol{\beta}_2 + \boldsymbol{\gamma}_2). \tag{5}$$

The given constraint translates into $\boldsymbol{\alpha}_2 = \mathbf{0}$, so the subsystem in Eq. 5 is completely immobilized; and the principle of virtual work requires that $\mathbf{S}_1^T\,\mathbf{v}_c = \mathbf{0}$, which implies $\boldsymbol{\gamma}_1 = \mathbf{0}$, so the subsystem in Eq. 4 is independent of the constraint. The equation of motion of the constrained system is therefore

$$\boldsymbol{\alpha}_1 = (\mathbf{S}_1^T \mathbf{M}\,\mathbf{S}_1)^{-1}\,\boldsymbol{\beta}_1 \tag{6}$$

in the coordinate system defined by $\mathbf{S}_1$ and $\mathbf{T}_1$, or

$$\mathbf{u} = \mathbf{S}_1\,(\mathbf{S}_1^T \mathbf{M}\,\mathbf{S}_1)^{-1}\,\mathbf{S}_1^T\,\mathbf{v} \tag{7}$$

in the original basis-pair.

The matrix expression on the RHS of this equation is the unconstrained, or active, component of $\mathbf{M}^{-1}$: the component that maps $T_1$ to $S_1$, and that accounts for the dynamics of the constrained system. Eq. 3 provides us with an alternative expression for this component:

$$\mathbf{S}_1 (\mathbf{S}_1^T \mathbf{M} \mathbf{S}_1)^{-1} \mathbf{S}_1^T = \mathbf{M}^{-1} - \mathbf{M}^{-1} \mathbf{T}_2 (\mathbf{T}_2^T \mathbf{M}^{-1} \mathbf{T}_2)^{-1} \mathbf{T}_2^T \mathbf{M}^{-1}. \qquad (8)$$

In other words, the system matrix of a constrained dynamical system is the difference between the system matrix of the unconstrained system and the matrix describing the immobilized dynamics. (Note that this is a general result: there is no assumption at this stage that the physical system is a rigid-body system.)

We can use Eq. 8 to derive alternative expressions for Eqs. 6 and 7. Applied to Eq. 7, the result is

$$\mathbf{u} = (\mathbf{M}^{-1} - \mathbf{M}^{-1} \mathbf{T}_2 (\mathbf{T}_2^T \mathbf{M}^{-1} \mathbf{T}_2)^{-1} \mathbf{T}_2^T \mathbf{M}^{-1}) \mathbf{v}. \qquad (9)$$

An alternative expression for Eq. 6 can be obtained by premultiplying both sides of Eq. 9 by $\mathbf{T}_1^T$, resulting in

$$\boldsymbol{\alpha}_1 = \mathbf{T}_1^T (\mathbf{M}^{-1} - \mathbf{M}^{-1} \mathbf{T}_2 (\mathbf{T}_2^T \mathbf{M}^{-1} \mathbf{T}_2)^{-1} \mathbf{T}_2^T \mathbf{M}^{-1}) \mathbf{T}_1 \boldsymbol{\beta}_1;$$

but a more useful version can be obtained by applying a more general mapping. Consider the set of subspaces $T_1'$ that satisfy $T_1' \oplus T_2 = V$. For each such subspace, there is a unique matrix $\mathbf{T}_1'$ that satisfies both Range$(\mathbf{T}_1') = T_1'$ and

$\mathbf{S}_1^T \mathbf{T}'_1 = \mathbf{1}$; and any vector $\mathbf{v} = \mathbf{T}_1 \boldsymbol{\beta}_1 + \mathbf{T}_2 \boldsymbol{\beta}_2$ can be decomposed uniquely into

$$\mathbf{v} = \mathbf{T}'_1 \boldsymbol{\beta}_1 + \mathbf{T}_2 \boldsymbol{\beta}'_2.$$

If $T'_1 = T_1$ then $\mathbf{T}'_1 = \mathbf{T}_1$ and $\boldsymbol{\beta}'_2 = \boldsymbol{\beta}_2$, otherwise $\boldsymbol{\beta}'_2 \neq \boldsymbol{\beta}_2$. Substituting this expression for $\mathbf{v}$ into Eq. 9 and premultiplying both sides by ${\mathbf{T}'_1}^T$ produces

$$\boldsymbol{\alpha}_1 = {\mathbf{T}'_1}^T (\mathbf{M}^{-1} - \mathbf{M}^{-1} \mathbf{T}_2 (\mathbf{T}_2^T \mathbf{M}^{-1} \mathbf{T}_2)^{-1} \mathbf{T}_2^T \mathbf{M}^{-1}) \mathbf{T}'_1 \boldsymbol{\beta}_1. \qquad (10)$$

The point of this exercise is that there exist values of $\mathbf{T}'_1$ that are much easier to calculate than $\mathbf{T}_1$.

As we shall see in the next section, this equation leads to the CFA, whereas Eq. 6 leads to the Newton-Euler factorization of the mass matrix [10], which is characteristic of standard dynamics algorithms.

# 4 The Constraint Force Algorithm

The CFA, like many other dynamics algorithms, uses an inverse dynamics function to calculate all of the force terms that depend only on position and velocity variables and other known quantities. This is a standard technique, and involves calling the inverse dynamics function with the acceleration vector set to zero, and subtracting the result from the input force vector [11].

Having already compensated for these effects, we are free to simplify the system by ignoring gravity, setting all velocities to zero, and so on.

Given a system of $N$ independent rigid bodies at rest, the equation of motion of the system can be expressed as a set of equations

$$\mathbf{a}_i = \mathbf{I}_i^{-1}\,\mathbf{f}_i, \qquad i = 1 \ldots N, \tag{11}$$

where $\mathbf{a}_i \in \mathsf{M}^6$ is the spatial acceleration of body $i$, $\mathbf{f}_i \in \mathsf{F}^6$ is the spatial force applied to body $i$, and $\mathbf{I}_i : \mathsf{M}^6 \mapsto \mathsf{F}^6$ is the spatial inertia of body $i$. Explicit expressions for $\mathbf{a}_i$, $\mathbf{f}_i$ and $\mathbf{I}_i$, and various other spatial quantities, in terms of 3-D linear and angular vectors and matrices can be found in [9, 1, 10] and various other sources; but note that the vectors we use here require that the 3-D linear-component vector be placed either consistently above or consistently below the 3-D angular-component vector. This is at variance with the arrangement in [9] and certain other works, where the line-vector component (angular motion or linear force) is placed on top and the free-vector component below.

The appearance of the two spaces $\mathsf{M}^6$ and $\mathsf{F}^6$ is an important mathematical detail. They contain the motion-type and force-type spatial vectors, respectively, and they form a dual system $\langle \mathsf{M}^6, \mathsf{F}^6, \cdot \rangle$ with the reciprocal scalar product. This formal separation of vectors into distinct force and motion spaces is the first step in applying the CoB technique.

We avoid using the terms 'twist' and 'wrench' to describe these vectors because there is no need for them actually to be twists or wrenches. They could just as easily be vectors of generalized accelerations and forces. We use only the dual-system property, not any of the special properties of twists and wrenches.

The individual equations in Eq. 11 can be combined into a single composite equation that describes the whole system:

$$\mathbf{a} = \mathbf{I}^{-1}\mathbf{f} \tag{12}$$

where $\mathbf{f} = [\mathbf{f}_1^T, \ldots, \mathbf{f}_N^T]^T \in \mathsf{F}^{6N}$, $\mathbf{a} = [\mathbf{a}_1^T, \ldots, \mathbf{a}_N^T]^T \in \mathsf{M}^{6N}$ and $\mathbf{I} = \mathrm{diag}(\mathbf{I}_i) : \mathsf{M}^{6N} \mapsto \mathsf{F}^{6N}$. The spaces $\mathsf{M}^{6N}$ and $\mathsf{F}^{6N}$ are Cartesian products of $N$ lots of $\mathsf{M}^6$ and $\mathsf{F}^6$; and they form a dual system, $\langle \mathsf{M}^{6N}, \mathsf{F}^{6N}, \cdot \rangle$, which enables the CoB technique to be applied globally to the whole system.

The equation of motion of a robot mechanism can be obtained by subjecting the free-body system to a kinematic constraint that models the effects of the joints. Let the robot mechanism consist of $N$ movable links numbered $1 \ldots N$ from base to tip, one fixed base link numbered 0, and $N$ joints numbered $1 \ldots N$ such that joint $i$ connects from link $i-1$ to link $i$. Let $\mathbf{h}_i$ and $\ddot{\mathbf{q}}_i$ be the motion axis (or subspace) and acceleration variable of joint $i$, and let $\boldsymbol{\tau}_i$ be the active joint force at joint $i$ (after subtraction of the inverse-dynamics force). As all velocity terms are already accounted for, the bodies are taken to be at rest.

Each joint imposes an acceleration constraint of the form

$$\mathbf{a}_i - \mathbf{a}_{i-1} = \mathbf{h}_i \, \ddot{\mathbf{q}}_i,$$

and the active joint forces are related to the link forces by

$$\boldsymbol{\tau}_i = \mathbf{h}_i^T \, \mathbf{f}_{Ji} = \mathbf{h}_i^T \, ( \sum_{j=i}^{N} \mathbf{f}_j ),$$

where $\mathbf{f}_{Ji}$ is the total force transmitted from link $i-1$ to link $i$ through joint $i$ (so $\mathbf{f}_{Ji} = \mathbf{f}_i + \mathbf{f}_{Ji+1}$). These equations can be combined into two system-wide equations

$$\mathbf{P} \, \mathbf{a} = \mathbf{H} \, \ddot{\mathbf{q}}, \tag{13}$$

$$\boldsymbol{\tau} = \mathbf{H}^T \, \mathbf{P}^{-T} \, \mathbf{f}, \tag{14}$$

where $\mathbf{H} = \mathrm{diag}(\mathbf{h}_i) : \mathsf{M}^n \mapsto \mathsf{M}^{6N}$, $\ddot{\mathbf{q}} = [\ddot{\mathbf{q}}_1^T, \ldots, \ddot{\mathbf{q}}_N^T]^T \in \mathsf{M}^n$, $\boldsymbol{\tau} = [\boldsymbol{\tau}_1^T, \ldots, \boldsymbol{\tau}_N^T]^T \in \mathsf{F}^n$ and $n$ is the degree of freedom of the robot mechanism. The matrices $\mathbf{P} : \mathsf{M}^{6N} \mapsto \mathsf{M}^{6N}$ and $\mathbf{P}^{-T} : \mathsf{F}^{6N} \mapsto \mathsf{F}^{6N}$ serve to propagate information between links, and are given by

$$\mathbf{P} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \cdots & & \mathbf{0} \\ -\mathbf{1} & \mathbf{1} & & & \\ \mathbf{0} & & \ddots & & \vdots \\ \vdots & & & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{1} & \mathbf{1} \end{bmatrix}, \qquad \mathbf{P}^{-T} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \cdots & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & & \mathbf{1} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{1} \end{bmatrix}.$$

Each block is a $6 \times 6$ zero, identity or negative identity matrix. $\mathbf{P}$ corresponds to $\mathcal{P}^T$ in [1] (not $\mathcal{P}$, as might at first appear, because of the reversed element-numbering scheme) and $\mathbf{P}^{-1}$ to $\phi^*$ in [10]. The apparent differences between $\mathbf{P}$ and these other matrices is simply an artifact of our choice of coordinates: the equations in this section are expressed, for simplicity, in absolute coordinates. Transforms into link coordinates can easily be added at a later stage.

To apply the CoB technique to this system, use

$$\mathbf{S}_1 = \mathbf{P}^{-1}\mathbf{H}, \qquad \mathbf{T}_2 = \mathbf{P}^T\mathbf{H}^\perp,$$

where $\mathbf{H}^\perp = \text{diag}(\mathbf{h}_i^\perp)$ and the matrices $\mathbf{h}_i^\perp$ represent the spaces of possible constraint forces for each joint. The resulting equation of motion is

$$\mathbf{a} = \mathbf{P}^{-1}\mathbf{H}\,(\mathbf{H}^T\mathbf{P}^{-T}\mathbf{I}\,\mathbf{P}^{-1}\mathbf{H})^{-1}\,\mathbf{H}^T\,\mathbf{P}^{-T}\,\mathbf{f}$$

or, in terms of joint force and acceleration,

$$\ddot{\mathbf{q}} = (\mathbf{H}^T \mathbf{P}^{-T} \mathbf{I}\, \mathbf{P}^{-1} \mathbf{H})^{-1} \boldsymbol{\tau}. \qquad (15)$$

On comparing Eq. 15 with Eq. 3 in [10], it can be seen that $\mathbf{H}^T \mathbf{P}^{-T}\, \mathbf{I}\, \mathbf{P}^{-1}\, \mathbf{H}$ is simply the standard, Newton-Euler factorization of the joint-space mass matrix, which leads directly to the Newton-Euler algorithm for inverse dynamics, and indirectly to several $O(N^3)$ algorithms for forward dynamics, including the composite-rigid-body, or Walker-Orin algorithm. The CF factorization of the mass matrix appears in the alternative form of Eq. 15, which is obtained via Eq. 10:

$$\ddot{\mathbf{q}} = {\mathbf{T}'_1}^T\, (\mathbf{I}^{-1} - \mathbf{I}^{-1}\, \mathbf{T}_2\, (\mathbf{T}_2^T \mathbf{I}^{-1} \mathbf{T}_2)^{-1}\, \mathbf{T}_2^T\, \mathbf{I}^{-1})\, \mathbf{T}'_1\, \boldsymbol{\tau}, \qquad (16)$$

where, for computational reasons,

$$\mathbf{T}'_1 = \mathbf{P}^T\, \mathbf{I}_X\, \mathbf{H}\, (\mathbf{H}^T \mathbf{I}_X \mathbf{H})^{-1}$$

and $\mathbf{I}_X : \mathsf{M}^{6N} \mapsto \mathsf{F}^{6N}$ is an arbitrary, block-diagonal SPD mapping. It is permissible to equate $\mathbf{I}_X$ with $\mathbf{I}$; but a better choice for computational purposes is whichever mapping happens to be the identity matrix in the current basis-pair.

Although Eq. 16 is more complicated than Eq. 15, and involves more calculations, every step in the evaluation of Eq. 16 can be accomplished in $O(\log(N))$ time or better on a computer with $O(N)$ processors. The same is not true of Eq. 15. The details are explained in [1].

Equation 16 is the correct version of the CFA. On comparing it with Eq. 22 in [1], equating $\mathcal{W}$ in that equation with $\mathbf{H}^{\perp}$ here, the only difference is the appearance of $\mathcal{P}\mathcal{H}$ in that equation where $\mathbf{T}_1'$ appears here. This difference can be traced back to Eqs. 6 and 7 in [1], where the notion of orthogonal complements is used incorrectly (see [3]).

This version is dimensionally correct and invariant with respect to changes of basis (provided you remember to use the correct transformation rule for each quantity); and it works for any type of joint. It also works for floating bases, since a floating-base system can be simulated by a fixed-base system with a 6-DoF joint between the fixed base and the first moving body, which is the floating base. This approach to floating bases is easier and more efficient than the method described in [2]. Note that $\mathbf{h}_i^{\perp}$ has dimensions $6\times 0$ for a 6-DoF joint, which means that it contributes six rows and zero columns to $\mathbf{H}^{\perp}$, and all of the elements on those six rows are zero.

The CFA can be extended to allow short side branches by replacing the original set of $N$ independent rigid bodies with a set of $N$ independent 1-handle articulated bodies [9]. No part of the CFA needs to be changed, but two additional calculations are required:

1. After the inverse-dynamics calculation, but before the first step of CFA, run the articulated-body algorithm (ABA) in parallel on each of the articulated bodies, up to the point where all the articulated-body inertias have been calculated (*i.e.*, the end of step 1 in [9], or up to Eq. 39 or Eq. 46 in [12]).

2. After the last step of CFA, use the known spatial accelerations of the bodies on the main chain to finish off the ABA calculation.

If the lengths of the side chains are no greater than $O(\log N)$, then this additional calculation does not affect the asymptotic time complexity of the CFA. Indeed, the overall efficiency of this extended CFA, measured in terms of calculations per body, is actually slightly better than that of basic CFA because the ABA has a lower calculations-per-body count than CFA. Notice also that the extended CFA is capable of calculating the dynamics of a system containing $O(N \log N)$ bodies in $O(\log N)$ time on $O(N)$ processors, provided it has the right connectivity.

# 5 Conclusion

This paper has presented a new technique for analysing constrained rigid-body systems by means of a change of basis in a dual system of vector spaces, and a new formulation of the CFA using this technique. This new formulation is invariant, dimensionally-correct, and works for any type of joint, including the 6-DoF joint that is used to model floating-base systems. It also caters for short, $O(\log N)$-length side branches off the main chain, which are implemented using articulated-body equations. Finally, the new formulation shows that the relationship between the CFA and conventional dynamics algorithms is explained by an equation that can be paraphrased as follows: 'constrained dynamics = unconstrained dynamics − immobilized dynamics'. Conventional algorithms are derived from the LHS of this equation, and the CFA from the RHS.

# References

[1] A. Fijany, I. Sharf and G. M. T. D'Eleuterio, "Parallel $O(\log N)$ Algorithms for Computation of Manipulator Forward Dynamics," *IEEE Trans. Robotics & Automation*, vol. 11, no. 3, pp. 389–400, June 1995.

[2] A. Fijany and A. K. Bejczy, "A New Algorithmic Framework for Robot Dynamics Analysis with Application to Space Robots Dynamics Simulation," in *Proc. 8th Int. Conf. on Advanced Robotics*, 1997, pp. 799–805.

[3] J. Duffy, "The Fallacy of Modern Hybrid Control Theory that is Based on "Orthogonal Complements" of Twist and Wrench Spaces," *J. Robotic Systems*, vol. 7, no. 2, pp. 139–144, 1990.

[4] A. De Luca and C. Manes, "Modelling of Robots in Contact with a Dynamic Environment," *IEEE Trans. Robotics and Automation*, vol. 10, no. 4, pp. 542–548, 1994.

[5] J. M. Selig and P. R. McAree, "A Simple Approach to Invariant Hybrid Control," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1996, pp. 2238–2245.

[6] J. De Schutter, D. Torfs, H. Bruyninckx and S. Dutré, "Invariant Hybrid Force/Position Control of a Velocity Controlled Robot with Compliant End Effector Using Modal Decoupling," *Int. J. Robotics Research*, vol. 16, no. 3, pp. 340–356, 1997.

[7] H. Lipkin and J. Duffy, "Hybrid Twist and Wrench Control for a Robotic Manipulator," *Trans. ASME, J. Mechanisms, Transmissions & Automation in Design*, vol. 110, no. 2, pp. 138–144, June 1988.

[8] T. Patterson and H. Lipkin, "Duality of Constrained Elastic Manipulation," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1991, pp. 2820–2825.

[9] R. Featherstone, *Robot Dynamics Algorithms*, Boston/Dordrecht/Lancaster: Kluwer Academic Publishers, 1987.

[10] G. Rodriguez, A. Jain and K. Kreutz-Delgado, "A Spatial Operator Algebra for Manipulator Modelling and Control," *Int. J. Robotics Research*, vol. 10, no. 4, pp. 371–381, 1991.

[11] M. W. Walker and D. E. Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms," *Trans. ASME, J. Dynamic Systems, Measurement & Control*, vol. 104, pp. 205–211, Sept. 1982.

[12] R. Featherstone, "The Calculation of Robot Dynamics using Articulated-Body Inertias," *Int. J. Robotics Research*, vol. 2, no. 1, pp. 13–30, 1983.