# A Quick Tutorial on Multibody Dynamics

C. Karen Liu
Sumit Jain

School of Interactive Computing
Georgia Institute of Technology

Technical Report No: GIT-GVU-15-01-1

# Contents

# 1 Introduction

If you have not read the excellent SIGGRAPH course notes on physics-based animation by Witkin and Baraff, you can stop reading further right now. Go look for those notes at `http://www.cs.cmu.edu/~baraff/sigcourse/` and come back when you fully understand everything in those notes.

If you are still reading this document, you probably fit the following profile. You are a computer scientist with no mechanical engineering background and minimal training in physics in high school but you are seriously interested in physics-based character animation. You have read Witkin and Baraff's SIGGRAPH course notes a few times but don't know where to go from simulating rigid bodies to human figures. You have played with some commercial physics engines like ODE (Open Dynamic Engine), PhysX, Havok, or Bullet, but you wish to simulate human behaviors more interesting than ragdoll effects.

Physics-based character animation consists of two parts: simulation and control. This document focuses on the simulation part. It's quite likely that you do not need to understand how the underlying simulation works if your control algorithm is simple enough. However, complex human behaviors often require sophisticated controllers that exploit the dynamics of a multibody system. A good understanding of multibody dynamics is paramount for designing effective controllers.

There are many ways to learn multibody dynamics. Reading a textbook on this topic or taking a course from the mechanical engineering department will both do the job. However, if you only want to learn the minimal set of multibody dynamics necessary to jump start your research in physics-based character animation, this document might be what you are looking for. In particular, this document attempts to answer the following questions.

- I know how to derive the equations of motion for one rigid body and I have seen people use the following equations for articulated rigid bodies, but I don't know how they are derived.

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}$$

- I have seen the Euler-Lagrange equation in the following form before, but I don't know how it is related to the equations of motion above.

$$\frac{d}{dt}\left(\frac{\partial T_i}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial T_i}{\partial \mathbf{q}} - \mathbf{Q} = 0$$

- I use generalized coordinates to compute the control forces, how do I convert them to Cartesian forces such that I can use simulators like ODE, PhysX, or Bullet which represent rigid bodies in the maximal coordinates?

- I heard that inverse dynamics can be computed efficiently using a recursive formulation. How does that work?

## 2 Lagrangian Dynamics

Articulated human motions can be described by a set of dynamic equations of motion of multibody systems. Since the direct application of Newton's second law becomes difficult when a complex articulated rigid body system is considered, we use *Lagrange's equations* derived from *D'Alembert's principle* to describe the dynamics of motion. To simplify the math, let's temporarily imagine that the entire human skeleton consists of a collection of particles $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{n_p}\}$. Each particle, $\mathbf{r}_i$, is defined by Cartesian coordinates that describe the translation with respective to the world coordinates. We can represent $\mathbf{r}_i$ by a set of *generalized coordinates* that indicate the joint configuration of the human skeleton:

$$\mathbf{r}_i = \mathbf{r}_i(q_1, q_2, \ldots, q_{n_j}, t) \tag{1}$$

where $t$ is the time and $q_j$ is a joint degree of freedom (DOF) in the skeleton. Each $q_j$ is a function of time but we assume that $\mathbf{r}_i$ is not an explicit function of time.

The virtual displacement $\delta\mathbf{r}_i$ refers to an infinitesimal change in the system coordinates such that the constraint remains satisfied. In the context of human skeleton, the system coordinates are the generalized coordinates $q_j$ and the constraint manifold lies in the Cartesian space. The virtual displacement $\delta\mathbf{r}_i$ is a tangent vector to the constraint manifold at a fixed time, written as

$$\delta\mathbf{r}_i = \sum_j \frac{\partial \mathbf{r}_i}{\partial q_j}\delta q_j \tag{2}$$

We can now write the virtual work done by a force $\mathbf{f}_i$ acting on particle $\mathbf{r}_i$ as

$$\mathbf{f}_i \cdot \delta\mathbf{r}_i = \mathbf{f}_i \cdot \sum_j \frac{\partial \mathbf{r}_i}{\partial q_j}\delta q_j \equiv \sum_j Q_{ij}\delta q_j = \mathbf{Q}_i \cdot \delta\mathbf{q} \tag{3}$$

where $Q_{ij} = \left(\frac{\partial \mathbf{r}_i}{\partial q_j}\right)^T \mathbf{f}_i$ is defined as the component of the *generalized force* associated with coordinate $q_j$. In vector form, $\mathbf{Q}_i$ is the generalized force corresponding to the Cartesian force $\mathbf{f}_i$ with the relation $\mathbf{Q}_i = J_i^T \mathbf{f}_i$, where $J_i$ is the Jacobian matrix with the $j^{th}$ column defined as $\frac{\partial \mathbf{r}_i}{\partial q_j}$.

From D'Alembert's principle, we know that the sum of the differences between the forces acting on a system and the inertial force of the system along any virtual displacement consistent with the constraints of the system, is zero. Therefore, the virtual work at $\mathbf{r}_i$ can be written as

$$\delta W_i = \mathbf{f}_i \cdot \delta\mathbf{r}_i = \mu_i \ddot{\mathbf{r}}_i \cdot \delta\mathbf{r}_i = \sum_j \mu_i \ddot{\mathbf{r}}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j}\delta q_j \tag{4}$$

where $\mu_i$ is the infinitesimal mass associated with $\mathbf{r}_i$. The component of inertial force associated with $q_j$ can be written as

$$\mu_i \ddot{\mathbf{r}}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} = \frac{d}{dt}\left(\mu_i \dot{\mathbf{r}}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j}\right) - \mu_i \dot{\mathbf{r}}_i \cdot \frac{d}{dt}\left(\frac{\partial \mathbf{r}_i}{\partial q_j}\right) \tag{5}$$

Now let us consider the velocity of $\mathbf{r}_i$ in terms of joint velocity $\dot{q}_j$

$$\dot{\mathbf{r}}_i = \sum_j \frac{\partial \mathbf{r}_i}{\partial q_j} \dot{q}_j \tag{6}$$

from which we derive the following two identities:

$$\frac{\partial \dot{\mathbf{r}}_i}{\partial \dot{q}_j} = \frac{\partial \mathbf{r}_i}{\partial q_j} \tag{7}$$

$$\frac{\partial \dot{\mathbf{r}}_i}{\partial q_j} = \sum_k \frac{\partial^2 \mathbf{r}_i}{\partial q_j \partial q_k} \dot{q}_k = \frac{d}{dt} \frac{\partial \mathbf{r}_i}{\partial q_j} \tag{8}$$

Using these two identities, we rewrite Equation (5) as

$$\mu_i \ddot{\mathbf{r}}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} = \frac{d}{dt} \left( \frac{\partial}{\partial \dot{q}_j} \left( \frac{1}{2} \mu_i \dot{\mathbf{r}}_i^T \dot{\mathbf{r}}_i \right) \right) - \frac{\partial}{\partial q_j} \left( \frac{1}{2} \mu_i \dot{\mathbf{r}}_i^T \dot{\mathbf{r}}_i \right) \tag{9}$$

We can denote the kinetic energy of $\mathbf{r}_i$ as

$$T_i = \frac{1}{2} \mu \dot{\mathbf{r}}_i^T \dot{\mathbf{r}}_i, \tag{10}$$

and rewrite Equation (9) as

$$\mu_i \ddot{\mathbf{r}}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} = \frac{d}{dt} \left( \frac{\partial T_i}{\partial \dot{q}_j} \right) - \frac{\partial T_i}{\partial q_j} \tag{11}$$

Combining the definition of generalized force (Equation (3)), D'Alembert's principle (Equation (4)), and the generalized inertial force (Equation (11)), we arrive at the following equation:

$$\left( \frac{d}{dt} \left( \frac{\partial T_i}{\partial \dot{q}_j} \right) - \frac{\partial T_i}{\partial q_j} \right) \delta q_j = Q_{ij} \delta q_j \tag{12}$$

If the set of generalized coordinates $q_j$ is linearly independent, Equation (12) leads to *Lagrangian equation*:

$$\frac{d}{dt} \left( \frac{\partial T_i}{\partial \dot{q}_j} \right) - \frac{\partial T_i}{\partial q_j} - Q_{ij} = 0 \tag{13}$$

**Equations of Motion in Vector Form.** Equation (13) is the equation of motion for one generalized coordinate in a multibody system. We can combine $n_j$ scalar equations into the familiar vector form

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q} \tag{14}$$

where $M(\mathbf{q})$ is the mass matrix, $C(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis and centrifugal term of the equation of motion, and $\mathbf{Q}$ is the vector of generalized forces for all the degrees of freedom (DOFs) in the system. $M$ only depends on $\mathbf{q}$ and $C$ depends quadratically on $\dot{\mathbf{q}}$.

How do we derive $M$ and $C$ from Equation (13)? Let us go back to the velocity of one particle $\mathbf{r}_i$:

$$\dot{\mathbf{r}}_i = \sum_j \frac{\partial \mathbf{r}_i}{\partial q_j} \dot{q}_j = J_i(\mathbf{q})\dot{\mathbf{q}} \tag{15}$$

where $J_i$ denotes the Jacobian of $\mathbf{r}_i$. By summing up all the particles in the system, the kinetic energy of the system can then be expressed as

$$T = \sum_i T_i = \sum_i \frac{1}{2}\mu \dot{\mathbf{r}}_i^T \dot{\mathbf{r}}_i = \sum_i \frac{1}{2}\mu (J_i\dot{\mathbf{q}})^T (J_i\dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T (\sum_i \mu J_i^T J_i)\dot{\mathbf{q}} = \frac{1}{2}\dot{\mathbf{q}}^T M(\mathbf{q})\dot{\mathbf{q}} \tag{16}$$

where we define the mass matrix, $M(\mathbf{q}) = \sum_i \mu J_i^T J_i$, and will shortly show it is indeed the mass matrix in Equation (14).

From Equation (16), we can derive the derivative terms to construct the Lagrange's equation (Equation (13)):

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\mathbf{q}}} - \frac{\partial T}{\partial \mathbf{q}} = M\ddot{\mathbf{q}} + \dot{M}\dot{\mathbf{q}} - \frac{1}{2}\dot{\mathbf{q}}^T \left(\frac{\partial M}{\partial \mathbf{q}}\right)^T \dot{\mathbf{q}} \equiv M\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) \tag{17}$$

Comparing Equation (17) to Equation (14), we confirm that the mass matrix is identical in both equations. $C$ is the Coriolis and centrifugal term in Equation (14) and is defined as $C = \dot{M}\dot{\mathbf{q}} - \frac{1}{2}\left(\frac{\partial M}{\partial \mathbf{q}}\dot{\mathbf{q}}\right)^T \dot{\mathbf{q}}$.

**Note.** In the second term of $C$, we introduce tensor notation $\frac{\partial M}{\partial \mathbf{q}}$, which implies that the $j^{th}$ element of the tensor $\frac{\partial M}{\partial \mathbf{q}}$ is the matrix $\frac{\partial M}{\partial q_j}$. Note that, in general, the quantity with notation $\frac{\partial M}{\partial \mathbf{q}}\dot{\mathbf{q}}$ is ***not*** equal to $\dot{M}$. This is because, the $j^{th}$ column of the matrix $\frac{\partial M}{\partial \mathbf{q}}\dot{\mathbf{q}}$ is the vector $\frac{\partial M}{\partial q_j}\dot{\mathbf{q}}$ or $\sum_k \frac{\partial (M)_k}{\partial q_j}\dot{q}_k$, where the notation $(A)_j$ denotes the $j^{th}$ column of the matrix $A$. In contrast, the $j^{th}$ column of the matrix $\dot{M}$ is $\sum_k \frac{\partial (M)_j}{\partial q_k}\dot{q}_k$.

Once we know how to compute the mass matrix, Coriolis and centrifugal terms, and generalized forces, we can compute the acceleration in generalized coordinates, $\ddot{\mathbf{q}}$, for *forward dynamics*. Conversely, if we are given $\ddot{\mathbf{q}}$ from a motion sequence, we can use these equations of motion to derive generalized forces for *inverse dynamics*.

The above formulation is convenient for a system consisting of finite number of mass points. However, for a dynamic system that consists of rigid bodies, there are infinitely many points contained in each rigid body making the above formulation intractable. In the following two sections, we view a rigid body as a continuum and derive compact equations of motions in both Cartesian coordinates and generalized coordinates.

# 3 Review: Newton-Euler equations

This section reviews Newton-Euler equations for rigid body dynamics. The derivation of mass matrix $M(\mathbf{q})$ and Coriolis and centrifugal term $C(\mathbf{q}, \dot{\mathbf{q}})$ for a rigid body will be presented in the next section. If you are familiar with Newton-Euler equations, you can skip this section and continue to the next. However, many math notations used in Witkin and Baraff's course notes are also reviewed in this section, such as linear momentum, angular momentum, skew-symmetric matrix and its properties.

To derive Newton-Euler equations, we begin with the momenta of the rigid body whose mass, position of the center of mass (COM), orientation, linear velocity of the COM, and angular velocity are $m$, $\mathbf{x}$, $R$, $\mathbf{v}$, and $\boldsymbol{\omega}$ respectively (these definitions are the same as are found in Witkin and Baraff's course notes). The linear momentum $\mathbf{P}$ is computed as:

$$\begin{aligned} \mathbf{P} &= \sum_i \mathbf{P}_i = \sum_i \mu \dot{\mathbf{r}}_i = \sum_i \mu(\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}'_i) \\ &= m\mathbf{v} \end{aligned} \tag{18}$$

where $\mathbf{r}'_i = \mathbf{r}_i - \mathbf{x}$. Because $\sum_i \mu \mathbf{r}'_i = \mathbf{0}$ (property of the COM), the second term vanishes. The angular momentum $\mathbf{L}$ about the COM is computed as:

$$\begin{aligned} \mathbf{L} &= \sum_i \mathbf{L}_i = \sum_i \mathbf{r}'_i \times \mathbf{P}_i \\ &= \sum_i \mu \mathbf{r}'_i \times (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}'_i) \\ &= \mathbf{0} + \sum_i \mu [\mathbf{r}'_i][\boldsymbol{\omega}]\mathbf{r}'_i = \left( \sum_i -\mu [\mathbf{r}'_i][\mathbf{r}'_i] \right) \boldsymbol{\omega} \end{aligned} \tag{19}$$

The notation $[\mathbf{a}]\mathbf{b}$ denotes the cross product $\mathbf{a} \times \mathbf{b}$ with $[\mathbf{a}]$ being the skew-symmetric matrix corresponding to the vector $\mathbf{a}$:

$$[\mathbf{a}] = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \tag{20}$$

Therefore the following identities hold: $[\mathbf{a}]\mathbf{b} = -[\mathbf{b}]\mathbf{a}$ and $[\mathbf{a}]^T = -[\mathbf{a}]$.

Now recall the inertia tensor about the COM defined in Witkin and Baraff's course notes: $I_c = \sum_i \mu((\mathbf{r}'^T_i \mathbf{r}'_i)\mathbf{I}_3 - \mathbf{r}'_i \mathbf{r}'^T_i)$, where $\mathbf{I}_3$ is the $3 \times 3$ identity matrix. We can easily show that $I_c = \sum_i -\mu[\mathbf{r}'_i][\mathbf{r}'_i]$ by verifying the identity $-[\mathbf{a}][\mathbf{a}] = (\mathbf{a}^T\mathbf{a})\mathbf{I}_3 - \mathbf{a}\mathbf{a}^T$. As a result, we write the angular momentum of a rigid body as:

$$\mathbf{L} = I_c \boldsymbol{\omega} \tag{21}$$

where the inertia tensor can be written as $I_c = R I_0 R^T$. $R$ is the rotation matrix corresponding to the orientation of the body and $I_0$ is the constant inertia tensor defined at zero rotation. From Witkin and Baraff's course notes, we also learned that the angular velocity in the skew-symmetric form is related to the rotation matrix $R$ as $[\boldsymbol{\omega}] = \dot{R}R^T$.

With these definitions, we can derive the equations of motion for a rigid body. The equations corresponding to the linear force can be evaluated as:

$$\mathbf{f} = \dot{\mathbf{p}} = m\dot{\mathbf{v}} \tag{22}$$

The equations corresponding to the torque can be evaluated as:

$$\begin{aligned}
\boldsymbol{\tau} &= \dot{\mathbf{L}} = (I_c\dot{\boldsymbol{\omega}}) \\
&= I_c\dot{\boldsymbol{\omega}} + (R\dot{I_0}R^T)\boldsymbol{\omega} = I_c\dot{\boldsymbol{\omega}} + \dot{R}I_0R^T\boldsymbol{\omega} + RI_0\dot{R}^T\boldsymbol{\omega} \\
&= I_c\dot{\boldsymbol{\omega}} + \dot{R}R^TI_c\boldsymbol{\omega} + I_c(\dot{R}R^T)^T\boldsymbol{\omega} \\
&= I_c\dot{\boldsymbol{\omega}} + [\boldsymbol{\omega}]I_c\boldsymbol{\omega} - I_c[\boldsymbol{\omega}]\boldsymbol{\omega} \quad \text{(Using the identity } [\boldsymbol{\omega}]^T = -[\boldsymbol{\omega}]) \\
&= I_c\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times I_c\boldsymbol{\omega}
\end{aligned} \tag{23}$$

Combining Equation (22) and Equation (23), we arrive at the Newton-Euler equations:

$$\begin{pmatrix} m\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & I_c \end{pmatrix} \begin{pmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\omega} \times I_c\boldsymbol{\omega} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{pmatrix} \tag{24}$$

# 4 Rigid Body Dynamics: Lagrange's equations

The Newton-Euler equations are defined in terms of velocities instead of position and orientation. We now derive the equations in generalized coordinates $\mathbf{q}$ that define the position and orientation. The first three coordinates are the same as the position of COM. The next three represent the rotation of the rigid body such as an exponential map or three Euler angles (or four coordinates can be used for a quaternion). In particular, we will show how mass matrix and Coriolis and centrifugal term are computed in Equation (14).

We start by computing the kinetic energy of the rigid body using the notions in Equation (18):

$$\begin{aligned} T &= \sum_i T_i = \sum_i \frac{1}{2}\mu \dot{\mathbf{r}}_i^T \dot{\mathbf{r}}_i = \sum_i \frac{1}{2}\mu(\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}'_i)^T(\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}'_i) \\ &= \sum_i \frac{1}{2}\mu(\mathbf{v}^T\mathbf{v} + \mathbf{v}^T[\boldsymbol{\omega}]\mathbf{r}'_i + \mathbf{r}'^T_i[\boldsymbol{\omega}]^T\mathbf{v} + \mathbf{r}'^T_i[\boldsymbol{\omega}]^T[\boldsymbol{\omega}]\mathbf{r}'_i) \end{aligned} \tag{25}$$

Because $\sum_i \mu \mathbf{r}'_i = \mathbf{0}$, the second term and the third term in Equation (25) vanish. Using the identity $[\boldsymbol{\omega}]\mathbf{r}'_i = -[\mathbf{r}'_i]\boldsymbol{\omega}$, we can rewrite Equation (25) as:

$$\begin{aligned} T &= \frac{1}{2}m\mathbf{v}^T\mathbf{v} + \frac{1}{2}\boldsymbol{\omega}^T\left(\sum_i -\mu[\mathbf{r}'_i][\mathbf{r}'_i]\right)\boldsymbol{\omega} \\ &= \frac{1}{2}m\mathbf{v}^T\mathbf{v} + \frac{1}{2}\boldsymbol{\omega}^T I_c \boldsymbol{\omega} \end{aligned} \tag{26}$$

The kinetic energy of a rigid body can be written in its vector form:

$$T = \frac{1}{2}(\mathbf{v}^T \;\; \boldsymbol{\omega}^T)\begin{pmatrix} m\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & I_c \end{pmatrix}\begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{pmatrix} \equiv \frac{1}{2}\mathbf{V}^T M_c \mathbf{V} \tag{27}$$

where $\mathbf{V} = (\mathbf{v}^T, \boldsymbol{\omega}^T)^T$, $M_c = \text{blockdiag}(m\mathbf{I}_3, I_c)$. We now relate the velocities in the Cartesian space $\mathbf{V}$ to the generalized velocities $\dot{\mathbf{q}}$. Let $\mathbf{x}(\mathbf{q})$ and $R(\mathbf{q})$ represent the position of the COM and the rotation matrix of the rigid body. The linear velocity of the COM is computed as:

$$\mathbf{v} = \dot{\mathbf{x}}(\mathbf{q}) = \frac{\partial \mathbf{x}}{\partial \mathbf{q}}\dot{\mathbf{q}} \equiv J_v \dot{\mathbf{q}} \tag{28}$$

The angular velocity is computed as:

$$\begin{aligned} [\boldsymbol{\omega}] &= \dot{R}(\mathbf{q})R^T(\mathbf{q}) \\ &= \sum_j \frac{\partial R}{\partial q_j} R^T \dot{q}_j \equiv \sum_j [\mathbf{j}_j]\dot{q}_j \end{aligned} \tag{29}$$

$\frac{\partial R}{\partial q_j} R^T$ is always a skew-symmetric matrix that we represent as $[\mathbf{j}_j]$ (skew-symmetric form of the vector $\mathbf{j}_j$). $\boldsymbol{\omega}$ can now be represented in the vector form as:

$$\boldsymbol{\omega} = J_\omega \dot{\mathbf{q}} \tag{30}$$

where $\mathbf{j}_j$ is the $j^{th}$ column of the matrix $J_\omega$.

Using Equation (28) and Equation (30), we can write:

$$\mathbf{V} = \begin{pmatrix} J_v \\ J_\omega \end{pmatrix} \dot{\mathbf{q}} \equiv J(\mathbf{q})\dot{\mathbf{q}} \tag{31}$$

Substituting the above in Equation (27), we get:

$$T = \frac{1}{2}\dot{\mathbf{q}}^T J^T M_c J \dot{\mathbf{q}} \tag{32}$$

Using the recipe for Lagrangian dynamics in Equation (13), we first compute $\frac{\partial T}{\partial \dot{q}_j}$ as:

$$\begin{aligned} \frac{\partial T}{\partial \dot{q}_j} &= \frac{1}{2}\dot{\mathbf{q}}^T J^T M_c (J)_j + \frac{1}{2}(J)_j^T M_c J\dot{\mathbf{q}} \\ &= (J)_j^T M_c J \dot{\mathbf{q}} \end{aligned} \tag{33}$$

where the notation $(A)_j$ denotes the $j^{th}$ column of the matrix A. The term $\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right)$ is computed as:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right) = (J)_j^T M_c J \ddot{\mathbf{q}} + (J)_j^T M_c \dot{J} \dot{\mathbf{q}} + (J)_j^T \dot{M}_c J \dot{\mathbf{q}} + (\dot{J})_j^T M_c J \dot{\mathbf{q}} \tag{34}$$

Now we evaluate the term $\frac{\partial T}{\partial q_j}$:

$$\begin{aligned} \frac{\partial T}{\partial q_j} &= \frac{1}{2}\dot{\mathbf{q}}^T J^T M_c \frac{\partial J}{\partial q_j}\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T J^T \frac{\partial M_c}{\partial q_j} J\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T \frac{\partial J^T}{\partial q_j} M_c J \dot{\mathbf{q}} \\ &= \dot{\mathbf{q}}^T \frac{\partial J^T}{\partial q_j} M_c J\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T J^T \frac{\partial M_c}{\partial q_j} J \dot{\mathbf{q}} \end{aligned} \tag{35}$$

Using the above equations, we write:

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right) - \frac{\partial T}{\partial q_j} &= (J)_j^T M_c J \ddot{\mathbf{q}} + (J)_j^T M_c \dot{J}\dot{\mathbf{q}} + (J)_j^T \dot{M}_c J \dot{\mathbf{q}} - \frac{1}{2}\dot{\mathbf{q}}^T J^T \frac{\partial M_c}{\partial q_j} J \dot{\mathbf{q}} \\ &\quad + \left( (\dot{J})_j^T M_c J \dot{\mathbf{q}} - \left(\frac{\partial J}{\partial q_j}\dot{\mathbf{q}}\right)^T M_c J \dot{\mathbf{q}} \right) \end{aligned} \tag{36}$$

Comparing Equation (36) to Equation (17), it seems that we can view the first term as the mass matrix multiplying by $\ddot{\mathbf{q}}$ and the rest terms as Coriolis and centrifugal forces. However, we will show that the third, fourth, and fifth terms of this equation can be greatly reduced.

**Third term:**

$$\begin{aligned} (J)_j^T \dot{M}_c J \dot{\mathbf{q}} &= (J_\omega)_j^T \dot{I}_c J_\omega \dot{\mathbf{q}} \quad \text{(The linear term in } M_c \text{ is constant: see Equation (27))} \\ &= \mathbf{j}_j^T (R I_0 \dot{R^T}) \boldsymbol{\omega} \quad (\mathbf{j}_j \text{ represents the } j^{th} \text{ column of } J_\omega\text{: see Equation (29))} \\ \text{term 3} &= \mathbf{j}_j^T [\boldsymbol{\omega}] I_c \boldsymbol{\omega} \quad \text{(From Equation (23))} \end{aligned} \tag{37}$$

**Fourth term:** The fourth term in Equation (36) can be simplified as:

$$\begin{aligned}
\frac{1}{2}\dot{\mathbf{q}}^T J^T \frac{\partial M_c}{\partial q_j} J\dot{\mathbf{q}} &= \frac{1}{2}(J_\omega \dot{\mathbf{q}})^T \frac{\partial I_c}{\partial q_j} J_\omega \dot{\mathbf{q}} \\
&= \frac{1}{2}\boldsymbol{\omega}^T \left( \frac{\partial R}{\partial q_j} I_0 R^T + R I_0 \frac{\partial R^T}{\partial q_j} \right) \boldsymbol{\omega} = \boldsymbol{\omega}^T \left( \frac{\partial R}{\partial q_j} I_0 R^T \right) \boldsymbol{\omega} \\
&= \boldsymbol{\omega}^T \left( \frac{\partial R}{\partial q_j} R^T I_c \right) \boldsymbol{\omega} \\
&= \boldsymbol{\omega}^T [\mathbf{j}_j] I_c \boldsymbol{\omega} \quad \text{(From Equation (29))} \\
\text{term 4} &= -\mathbf{j}_j^T [\boldsymbol{\omega}] I_c \boldsymbol{\omega} \quad \text{(Using the identity } \mathbf{a}\cdot(\mathbf{b}\times\mathbf{c}) = -\mathbf{b}\cdot(\mathbf{a}\times\mathbf{c})\text{)} \qquad (38)
\end{aligned}$$

**Fifth term:** To simplify the fifth term in Equation (36), we explicitly express it using the linear and angular components:

$$\begin{pmatrix} (\dot{J_v})_j^T & (\dot{J_\omega})_j^T \end{pmatrix} \begin{pmatrix} m\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & I_c \end{pmatrix} \begin{pmatrix} J_v \dot{\mathbf{q}} \\ J_\omega \dot{\mathbf{q}} \end{pmatrix} - \begin{pmatrix} \left(\frac{\partial J_v}{\partial q_j}\dot{\mathbf{q}}\right)^T & \left(\frac{\partial J_\omega}{\partial q_j}\dot{\mathbf{q}}\right)^T \end{pmatrix} \begin{pmatrix} m\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & I_c \end{pmatrix} \begin{pmatrix} J_v \dot{\mathbf{q}} \\ J_\omega \dot{\mathbf{q}} \end{pmatrix} \quad (39)$$

The linear term can be extracted and simplified as:

$$\begin{aligned}
m\left( (\dot{J_v})_j - \left( \frac{\partial J_v}{\partial q_j}\dot{\mathbf{q}} \right) \right)^T J_v \dot{\mathbf{q}} &= m \left( \sum_k \frac{\partial (J_v)_j}{\partial q_k}\dot{q}_k - \sum_k \frac{\partial (J_v)_k}{\partial q_j}\dot{q}_k \right)^T J_v \dot{\mathbf{q}} \\
&= m \left( \sum_k \frac{\partial^2 \mathbf{x}}{\partial q_j \partial q_k}\dot{q}_k - \sum_k \frac{\partial^2 \mathbf{x}}{\partial q_k \partial q_j}\dot{q}_k \right)^T J_v \dot{\mathbf{q}} \\
\text{term 5 (linear)} &= 0 \qquad (40)
\end{aligned}$$

The above derivation uses the property of the Jacobian of the linear velocity $(J_v)_j = \frac{\partial \mathbf{x}}{\partial q_j} \ \forall j$ (See Equation (28)).

We now extract and simplify the angular term in Equation (39) as:

$$\begin{aligned}
\left( (\dot{J_\omega})_j - \left( \frac{\partial J_\omega}{\partial q_j}\dot{\mathbf{q}} \right) \right)^T I_c J_\omega \dot{\mathbf{q}} &= \left( \sum_k \frac{\partial \mathbf{j}_j}{\partial q_k}\dot{q}_k - \sum_k \frac{\partial \mathbf{j}_k}{\partial q_j}\dot{q}_k \right)^T I_c \boldsymbol{\omega} \\
&= \left( \sum_k \left( \frac{\partial \mathbf{j}_j}{\partial q_k} - \frac{\partial \mathbf{j}_k}{\partial q_j} \right) \dot{q}_k \right)^T I_c \boldsymbol{\omega} \equiv \left( \sum_k \mathbf{z}_{jk}\dot{q}_k \right)^T I_c \boldsymbol{\omega} \quad (41)
\end{aligned}$$

Now let us evaluate the term denoted by $\mathbf{z}_{jk}$. Consider the skew symmetric form:

$$
\begin{aligned}
[\mathbf{z}_{jk}] &= \left[\frac{\partial \mathbf{j}_j}{\partial q_k} - \frac{\partial \mathbf{j}_k}{\partial q_j}\right] = \frac{\partial [\mathbf{j}_j]}{\partial q_k} - \frac{\partial [\mathbf{j}_k]}{\partial q_j} \quad \text{(Using linearity of the skew symmetric matrix)} \\
&= \left(\frac{\partial^2 R}{\partial q_j \partial q_k} R^T + \frac{\partial R}{\partial q_j}\frac{\partial R^T}{\partial q_k}\right) - \left(\frac{\partial^2 R}{\partial q_k \partial q_j} R^T + \frac{\partial R}{\partial q_k}\frac{\partial R^T}{\partial q_j}\right) \quad \text{(From Equation (29))} \\
&= \frac{\partial R}{\partial q_j} R^T \left(\frac{\partial R}{\partial q_k} R^T\right)^T - \frac{\partial R}{\partial q_k} R^T \left(\frac{\partial R}{\partial q_j} R^T\right)^T \\
&= -[\mathbf{j}_j][\mathbf{j}_k] + [\mathbf{j}_k][\mathbf{j}_j] \quad \text{(Using the identity } [\mathbf{a}]^T = -[\mathbf{a}]) \\
&= [\mathbf{j}_k \times \mathbf{j}_j] \quad \text{(Using the identity } [\mathbf{a} \times \mathbf{b}] = [\mathbf{a}][\mathbf{b}] - [\mathbf{b}][\mathbf{a}]) \\
\Rightarrow \mathbf{z}_{jk} &= \mathbf{j}_k \times \mathbf{j}_j = [\mathbf{j}_k]\mathbf{j}_j
\end{aligned} \tag{42}
$$

Substituting the above in Equation (41), we get:

$$
\begin{aligned}
\left(\sum_k \mathbf{z}_{jk}\dot{q}_k\right)^T I_c \boldsymbol{\omega} &= \left(\sum_k [\mathbf{j}_k]\mathbf{j}_j \dot{q}_k\right)^T I_c \boldsymbol{\omega} \\
&= \left(\left(\sum_k [\mathbf{j}_k]\dot{q}_k\right)\mathbf{j}_j\right)^T I_c \boldsymbol{\omega} \\
&= \left(\left[\sum_k \mathbf{j}_k \dot{q}_k\right]\mathbf{j}_j\right)^T I_c \boldsymbol{\omega} \\
&= ([J_\omega \dot{\mathbf{q}}]\mathbf{j}_j)^T I_c \boldsymbol{\omega} = ([\boldsymbol{\omega}]\mathbf{j}_j)^T I_c \boldsymbol{\omega} \\
\text{term 5 (angular)} &= -\mathbf{j}_j^T [\boldsymbol{\omega}] I_c \boldsymbol{\omega}
\end{aligned} \tag{43}
$$

**Put it together:** Finally, we substitute the terms computed in Equation (37), Equation (38), Equation (40) and Equation (43) into Equation (36) and rewrite it as:

$$
\begin{aligned}
\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right) - \frac{\partial T}{\partial q_j} &= (J)_j^T M_c J \ddot{\mathbf{q}} + (J)_j^T M_c \dot{J} \dot{\mathbf{q}} + \mathbf{j}_j^T [\boldsymbol{\omega}] I_c \boldsymbol{\omega} \\
&= \left((J)_j^T M_c J\right) \ddot{\mathbf{q}} + \left((J)_j^T M_c \dot{J} + (J)_j^T [\tilde{\boldsymbol{\omega}}] M_c J\right) \dot{\mathbf{q}} \\
\text{where } [\tilde{\boldsymbol{\omega}}] &= \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & [J_\omega \dot{\mathbf{q}}] \end{pmatrix}
\end{aligned} \tag{44}
$$

Writing the equations for all the $q_j$ in the vector form, we get:

$$
\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial T}{\partial \mathbf{q}} = \left(J^T M_c J\right) \ddot{\mathbf{q}} + \left(J^T M_c \dot{J} + J^T [\tilde{\boldsymbol{\omega}}] M_c J\right) \dot{\mathbf{q}} \tag{45}
$$

Note that the second term in the above equation involves the computation of $\dot{J}$ that can be computed as $\sum_k \frac{\partial J}{\partial q_k}\dot{q}_k$. In other words, we will need to compute the first and the second derivatives of a rotation matrix (i.e. $\frac{\partial R}{\partial q_j}$ and $\frac{\partial^2 R}{\partial q_i \partial q_k}$) in order to compose Jacobian $J$ and its time derivative $\dot{J}$ in Equation (45).

**Derivation using Newton-Euler equations.** We can alternatively derive the result in Equation (45) from the Newton-Euler equations in Equation (24). Using Equation (31), we substitute the Cartesian velocities $\mathbf{v}, \boldsymbol{\omega}$ in terms of the generalized velocities $\dot{\mathbf{q}}$ into Equation (24) and get:

$$\begin{aligned} M_c(\dot{J}\dot{\mathbf{q}}) + \begin{pmatrix} \mathbf{0} \\ (J_\omega \dot{\mathbf{q}}) \times I_c J_\omega \dot{\mathbf{q}} \end{pmatrix} &= \begin{pmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{pmatrix} \\ \Rightarrow M_c J \ddot{\mathbf{q}} + M_c \dot{J} \dot{\mathbf{q}} + [\tilde{\boldsymbol{\omega}}] M_c J \dot{\mathbf{q}} &= \begin{pmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{pmatrix} \end{aligned} \tag{46}$$

From the principle of virtual work in Equation (3), we convert the Cartesian-space forces to the Generalized space by pre-multiplying the above equation with the transpose of the Jacobian $J$:

$$\left(J^T M_c J\right) \ddot{\mathbf{q}} + \left(J^T M_c \dot{J} + J^T [\tilde{\boldsymbol{\omega}}] M_c J\right) \dot{\mathbf{q}} \quad = \quad J_v^T \mathbf{f} + J_\omega^T \boldsymbol{\tau} \tag{47}$$

The LHS of Equation (47) is identical to the RHS of Equation (45) and they are of the form $M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}$, where the Mass matrix, the Coriolis term and the generalized forces are defined as:

$$\begin{aligned} M(\mathbf{q}) &= J^T M_c J \\ C(\mathbf{q}, \dot{\mathbf{q}}) &= (J^T M_c \dot{J} + J^T [\tilde{\boldsymbol{\omega}}] M_c J) \dot{\mathbf{q}} \\ \mathbf{Q} &= J_v^T \mathbf{f} + J_\omega^T \boldsymbol{\tau} \end{aligned} \tag{48}$$

# 5 Articulated Rigid Body Dynamics

We now derive the equations of motion for an articulated rigid body structure. We follow the derivation of rigid body dynamics in generalized coordinates from Section 4.

An articulated rigid body system is represented as a set of rigid bodies connected through joints in a tree structure. Every rigid link has exactly one *parent* joint. The joint corresponding to the root of the tree is special; the root link does not link to any other rigid link. The generalized coordinates are therefore the DOFs of the root link of the tree (that may represent the global translation and rotation) and the joint angles corresponding to the admissible joint rotations for all the other joints.

## 5.1 Definitions

The state of an articulated rigid body system can be expressed as $(\mathbf{x}_k, R_k, \mathbf{v}_k, \boldsymbol{\omega}_k)$, where $k = 1, \cdots, m$ and $m$ is the number of rigid links. Here $\mathbf{x}_k$ and $R_k$ are the position of the COM and the orientation of the rigid link $k$, and $(\mathbf{v}_k, \boldsymbol{\omega}_k)$ are the linear and angular velocity of the rigid link $k$ viewed in the world frame. Similarly, we define the Cartesian force and torque applied on rigid link $k$ as $(\mathbf{f}_k, \boldsymbol{\tau}_k)$, both of which are expressed in the world frame.

The same articulated rigid body system can be represented in generalized coordinates. We define the generalized state as $(\mathbf{q}, \dot{\mathbf{q}})$, where $\mathbf{q} = (\mathbf{q}_1, \ldots, \mathbf{q}_k, \ldots, \mathbf{q}_m)$ and each $\mathbf{q}_k$ is the set of DOFs of the joint that connects the link $k$ to its parent link (see Figure 1).

We list a few notations and definitions for an articulated rigid body system with $m$ rigid links.

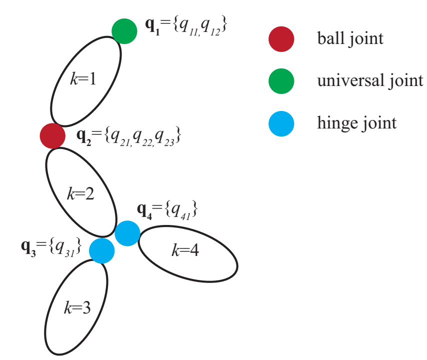


Figure 1: An articulated system.

- $p(k)$ returns the index of the parent link of link $k$. This is illustrated in Figure 1, $p(4) = 2$. $p(1, k)$ returns the indices of all the links in the chain from the root to the link $k$ (including $k$), e.g. $p(1, 4) = \{1, 2, 4\}$
- $n(k)$ returns the number of DOFs in the joint that connects the link $k$ to the parent link $p(k)$. For example in Figure 1, $n(2) = 3$, $n(3) = 1$ etc. We denote the total number of DOFs in the system by $n$. e.g. $n = 7$ in Figure 1.
- $R_k$ is the local rotation matrix for the link $k$ and depends only on the DOFs $\mathbf{q}_k$. $R_k^0$ is the chain of rotational transformations from the world frame to the local frame of the link $k$. Therefore, $R_k^0 = R_{p(k)}^0 R_k$. Since the link 1 does not have a parent link, $R_{p(1)}^0 = \mathbf{I}_3$.

## 5.2 Cartesian and generalized velocities

For a single rigid body, Equation (28) and Equation (30) describe the relation between the Cartesian velocities and the generalized velocities. For an articulated rigid body system, we use the same recipe as rigid body dynamics in Section 4 and define the Jacobians for each rigid link that relate its respective Cartesian velocities to the generalized velocity of the entire system.

We start with deriving the relation for the angular velocity. The angular velocity (in skew-symmetric matrix form) of link $k$ viewed in the world frame is:

$$\begin{aligned}
[\boldsymbol{\omega}_k] &= \dot{R}^0_k {R^0_k}^T = (\dot{\overbrace{R^0_{p(k)} R_k}})(R^0_{p(k)} R_k)^T \\
&= (\dot{R}^0_{p(k)} R_k + R^0_{p(k)} \dot{R}_k) R_k^T {R^0_{p(k)}}^T \\
&= \dot{R}^0_{p(k)} {R^0_{p(k)}}^T + R^0_{p(k)} \left(\dot{R}_k R_k^T\right) {R^0_{p(k)}}^T \equiv [\boldsymbol{\omega}_{p(k)}] + R^0_{p(k)} [\hat{\boldsymbol{\omega}}_k] {R^0_{p(k)}}^T
\end{aligned} \tag{49}$$

In the above equation, we define $[\hat{\boldsymbol{\omega}}_k] = \dot{R}_k R_k^T$ to denote the angular velocity of the link $k$ relative to its parent link $p(k)$, expressed in the frame of $p(k)$. We can further write $\hat{\boldsymbol{\omega}}_k = \hat{J}_{\omega k} \dot{\mathbf{q}}_k$ where $\hat{J}_{\omega k}$ is the *local* Jacobian matrix that relates the joint velocity of link $k$ to the relative angular velocity in the frame of $p(k)$. The dimension of $\hat{J}_{\omega k}$ is $3 \times n(k)$.

Using a property of skew symmetric matrices, $[R\boldsymbol{\omega}] = R[\boldsymbol{\omega}]R^T$, we can express Equation (49) in vector form as:

$$\begin{aligned}
\boldsymbol{\omega}_k &= \boldsymbol{\omega}_{p(k)} + R^0_{p(k)} \hat{J}_{\omega k} \dot{\mathbf{q}}_k \\
&= \sum_{l \in p(1,k)} R^0_{p(l)} \hat{J}_{\omega l} \dot{\mathbf{q}}_l \quad \text{(By unrolling the recursive definition)} \\
&\equiv J_{\omega k} \dot{\mathbf{q}}
\end{aligned} \tag{50}$$

where the Jacobian $J_{\omega k}$ is:

$$J_{\omega k} = \begin{pmatrix} \hat{J}_{\omega 1} & \dots & R^0_{p(l)} \hat{J}_{\omega l} & \dots & \mathbf{0} & \dots \end{pmatrix} \tag{51}$$

Note that the zero matrices $\mathbf{0}$ of size $3 \times n(l)$ in $J_{\omega k}$ correspond to joint DOFs $\mathbf{q}_l$ that are *not* in the chain of transformations from the root to the link $k$. Let us look at a couple of examples using the articulated rigid body system in Figure 1:

$$\begin{aligned}
\boldsymbol{\omega}_1 &= (\hat{J}_{\omega 1} \;\; \mathbf{0} \;\; \mathbf{0} \;\; \mathbf{0}) \dot{\mathbf{q}} \\
\boldsymbol{\omega}_4 &= (\hat{J}_{\omega 1} \;\; R^0_1 \hat{J}_{\omega 2} \;\; \mathbf{0} \;\; R^0_2 \hat{J}_{\omega 4}) \dot{\mathbf{q}}
\end{aligned}$$

where $\hat{J}_{\omega 1} \in \Re^{3 \times 2}$, $\hat{J}_{\omega 2} \in \Re^{3 \times 3}$ and $\hat{J}_{\omega 4} \in \Re^{3 \times 1}$. Depending on the representation of the rotation $\mathbf{q}_k$, $\hat{J}_{\omega k}$ can assume different values and dimensions. For example, if the joint between link 1 and link 2 in Figure 1 is represented as three Euler rotations, $R^{(x)}$, $R^{(y)}$, and $R^{(z)}$ such that $R_2(\mathbf{q}_2) = R^{(x)}(q_{21}) R^{(y)}(q_{22}) R^{(z)}(q_{23})$, we have:

$$\hat{J}_{\omega 2} = \begin{pmatrix} \begin{matrix} 1 \\ 0 \\ 0 \end{matrix} & R^{(x)} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} & R^{(x)} R^{(y)} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{pmatrix} \tag{52}$$

If the joint is represented as a quaternion or an exponential map, $\hat{J}_{\omega k}$ does not have a simple form. As an example, the relation between the rotation matrix $R_k$ and the exponential map representation $\mathbf{q}_k = (q_{k1}, q_{k2}, q_{k3})$ can be written as:

$$R_k(\mathbf{q}_k) = e^{[\mathbf{q}_k]} = \mathbf{I}_3 + \frac{sin\theta}{\theta}[\mathbf{q}_k] + \frac{1-cos\theta}{\theta^2}[\mathbf{q}_k]^2 \tag{53}$$

where $\theta = ||\mathbf{q}_k||$. The Jacobian $\hat{J}_{\omega k}$ can be derived by equating the result of $\dot{R}_k R_k^T$ to $[\hat{J}_{\omega k}\dot{\mathbf{q}}_k]$:

$$\begin{aligned}\hat{J}_{\omega k} &= R_k\left(\mathbf{I}_3 - \frac{1-cos\theta}{\theta^2}[\mathbf{q}_k] + \frac{\theta - sin\theta}{\theta^3}[\mathbf{q}_k]^2\right) \\ &= \mathbf{I}_3 + \frac{1-cos\theta}{\theta^2}[\mathbf{q}_k] + \frac{\theta - sin\theta}{\theta^3}[\mathbf{q}_k]^2 \end{aligned} \tag{54}$$

For the case when $\theta \to 0$, $R_k$ and $\hat{J}_{\omega k}$ can be approximated as follows:

$$R_k = \mathbf{I}_3 + [\mathbf{q}_k] + \frac{1}{2}[\mathbf{q}_k]^2 \tag{55}$$

$$\hat{J}_{\omega k} = \mathbf{I}_3 + \frac{1}{2}[\mathbf{q}_k] + \frac{1}{6}[\mathbf{q}_k]^2 \tag{56}$$

Similar to the angular velocity, the linear velocity of the center of mass of the link $k$ can be expressed in terms of the generalized velocity:

$$\mathbf{v}_k = J_{vk}\dot{\mathbf{q}}, \quad \text{where} \;\; J_{vk} = \frac{\partial \mathbf{x}_k}{\partial \mathbf{q}} = \frac{\partial W_k^0 \mathbf{c}_k}{\partial \mathbf{q}}. \tag{57}$$

where the chain of homogeneous transformations from the world frame to the local frame of link $k$ is denoted as $W_k^0$. Note that $W_k^0$ is different from $R_k^0$ in that $W_k^0$ includes the translational transformations. $\mathbf{c}_k$ is a constant vector that denotes the center of mass of link $k$ in its local frame.

We can concatenate the Cartesian velocities into a single vector $\mathbf{V}_k$ and denote the relation as:

$$\begin{aligned} \mathbf{V}_k &= J_k\dot{\mathbf{q}} \\ \text{where} \quad & \mathbf{V}_k = \begin{pmatrix} \mathbf{v}_k \\ \boldsymbol{\omega}_k \end{pmatrix} \text{ and } J_k = \begin{pmatrix} J_{vk} \\ J_{\omega k} \end{pmatrix} \end{aligned} \tag{58}$$

## 5.3 Equations of motion

We now derive the equations of motion of an articulated rigid body system in generalized coordinates. The kinetic energy $T$ of the entire system can be expressed as the sum of kinetic energies of all the rigid links as $T = \sum_k T_k$. Therefore the equations of motion of the system can be computed as:

$$
\begin{aligned}
\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial T}{\partial \mathbf{q}} &= \frac{d}{dt}\left(\frac{\partial \sum_k T_k}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \sum_k T_k}{\partial \mathbf{q}} \\
&= \sum_k \left(\frac{d}{dt}\left(\frac{\partial T_k}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial T_k}{\partial \mathbf{q}}\right) \\
&= \sum_k \left(\left(J_k^T M_{ck} J_k\right)\ddot{\mathbf{q}} + \left(J_k^T M_{ck} \dot{J}_k + J_k^T [\tilde{\boldsymbol{\omega}}_k] M_{ck} J_k\right)\dot{\mathbf{q}}\right) \\
&= \sum_k \left(J_k^T M_{ck} J_k\right)\ddot{\mathbf{q}} + \sum_k \left(J_k^T M_{ck} \dot{J}_k + J_k^T [\tilde{\boldsymbol{\omega}}_k] M_{ck} J_k\right)\dot{\mathbf{q}} \qquad (59)
\end{aligned}
$$

In deriving the above equation, we use the equations of motion in generalized coordinates for a single rigid body defined in Equation (45) subscripted by $k$ for the dynamics of $k^{th}$ link in the multibody system. The Jacobian $J_k$ for the $k^{th}$ link is defined in Equation (58).

# 6 Conversion between Cartesian and Generalized Coordinates

In practice, we often want to use third-party rigid body simulators rather than develop our own. There are a few widely used physics engines that provide efficient, robust, and fairly accurate rigid body simulation and collision handling. Open Dynamic Engine (ODE), PhysX, and Bullet are perhaps the most popular free choices among game developers and academic researchers. These commercial simulators use the maximal representation rather than generalized coordinates described above. That is, these simulators represent each link in the articulated rigid body system as six DOFs, leading to a redundant system with additional constraints between links. A common practice is to develop control algorithms in generalized coordinates and do forward simulation using a commercial physics engine, such as ODE. This requires some conversion between Cartesian and generalized coordinates.

## 6.1 Velocity conversion

We can concatenate all $2m$ Jacobian matrices corresponding to each link into a single Jacobian that relates the generalized velocity to the Cartesian velocity of each link:

$$\mathbf{V} \equiv \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_m \\ \boldsymbol{\omega}_1 \\ \vdots \\ \boldsymbol{\omega}_m \end{pmatrix} = \begin{pmatrix} J_{v1} \\ \vdots \\ J_{vm} \\ J_{\omega 1} \\ \vdots \\ J_{\omega m} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_1 \\ \vdots \\ \dot{\mathbf{q}}_m \end{pmatrix} \equiv \begin{pmatrix} J_v \\ J_\omega \end{pmatrix} \dot{\mathbf{q}} \equiv J\dot{\mathbf{q}} \tag{60}$$

Typically, the Jacobian $J$ is full column rank because the number of DOFs in the maximal representation is more than that in the generalized representation, i.e $6m > n$. To compute $\dot{\mathbf{q}}$ from $\mathbf{V}$, we will end up solving an over-constrained linear system. We can use the pseudo inverse of $J$ to compute $\dot{\mathbf{q}}$:

$$\dot{\mathbf{q}} = J^+\mathbf{V} \tag{61}$$

where the pseudo-inverse notation $J^+ = (J^TJ)^{-1}J^T$. If this least-squares solution does not exactly solve the linear system (i.e. $J\dot{\mathbf{q}} = \mathbf{V}$), it indicates that $\mathbf{V}$ cannot be achieved in the generalized coordinates without violating constraints of the system (e.g. constraints that keep links connected).

Computing $J^+$ may be expensive for a system with many rigid links. Alternatively, we can rewrite the equation using the relative velocity between a child and a parent link expressed in the local frame of the parent, instead of using velocities of each link expressed in the world frame. As an example, we write the simplified expression for the angular velocity of link $k$

using Equation (50) as:

$$
\begin{aligned}
\boldsymbol{\omega}_k - \boldsymbol{\omega}_{p(k)} &= R^0_{p(k)} \hat{\boldsymbol{\omega}}_k = R^0_{p(k)} \hat{J}_{\omega k} \dot{\mathbf{q}}_k \\
\Rightarrow (-\mathbf{I}_3 \;\; \mathbf{I}_3) \begin{pmatrix} \boldsymbol{\omega}_{p(k)} \\ \boldsymbol{\omega}_k \end{pmatrix} &= R^0_{p(k)} \hat{J}_{\omega k} \dot{\mathbf{q}}_k \qquad (62)
\end{aligned}
$$

Combining these equations for all the links, we get:

$$
\begin{aligned}
D\boldsymbol{\omega} = DJ_\omega \dot{\mathbf{q}} &= \text{blockdiag}(\hat{J}_{\omega 1}, \ldots, R^0_{p(m)} \hat{J}_{\omega m}) \dot{\mathbf{q}} \\
&= \text{blockdiag}(\mathbf{I}_3, \ldots, R^0_{p(m)}) \text{blockdiag}(\hat{J}_{\omega 1}, \ldots, \hat{J}_{\omega m}) \dot{\mathbf{q}} \\
&\equiv R \hat{J}_\omega \dot{\mathbf{q}} \qquad (63)
\end{aligned}
$$

where $D$ is a constant matrix that encodes the connectivity between links. For example, matrix $D$ for the system in Figure 1 looks like:

$$
D = \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_3 & \mathbf{0} & \mathbf{I}_3 \end{pmatrix} \qquad (64)
$$

The relations between $\hat{\boldsymbol{\omega}}$ and $\boldsymbol{\omega}$, and $\hat{J}_\omega$ and $J_\omega$ follow from Equation (63):

$$
\begin{aligned}
\hat{\boldsymbol{\omega}} &= R^T D \boldsymbol{\omega} \\
\text{and } \hat{J}_\omega &= R^T D J_\omega \qquad (65)
\end{aligned}
$$

The matrix $\hat{J}_\omega$ being block diagonal is much sparser as compared to $J_\omega$.

If $\dot{\mathbf{q}}$ satisfies the over-constrained system of equations $\mathbf{V} = J\dot{\mathbf{q}}$, using any $n$ independent constraints out of $6m$ to solve this linear system will result in the same $\dot{\mathbf{q}}$. This can be explained by the problem of fitting an unknown plane to $6m$ 3D points as $A\mathbf{x} = \mathbf{b}$, where $A \in \Re^{6m \times 3}$. If all $6m$ points happen to lie on a plane, i.e. there exists an $\mathbf{x}$ that exactly satisfies the over-constrained system, any three distinctive points we pick as the constraints will result in the same plane. Therefore, if we know $\mathbf{V}$ can be achieved in the generalized coordinates, we can pick a subset of rows from $J$ to form a $J'$ such that the rank of $J'$ is $n$, and compute $\dot{\mathbf{q}} = J'^{+}\mathbf{V}'$, where $\mathbf{V}'$ are the velocity components corresponding to the rows in $J'$. The solution $\dot{\mathbf{q}}$ to this system will be the same for any $J'$.

For a system with only rotational DOFs, it is sufficient to invert only $J_\omega$ which is also a full column rank matrix ($J_\omega \in \Re^{3m \times n}$ and $n \leq 3m$). This is because each rotational joint can have at most three independent DOFs. We then can compute the velocities of the rotational DOFs $\dot{\mathbf{q}}$ as:

$$
\begin{aligned}
\dot{\mathbf{q}} &= J^+_\omega \boldsymbol{\omega} \\
&= \hat{J}^+_\omega R^T D \boldsymbol{\omega} = \hat{J}^+_\omega \hat{\boldsymbol{\omega}} \\
&= \text{blockdiag}\left(\hat{J}^+_{\omega 1}, \ldots, \hat{J}^+_{\omega m}\right) \hat{\boldsymbol{\omega}} \\
\text{or } \dot{\mathbf{q}}_k &= \hat{J}^+_{\omega k} \hat{\boldsymbol{\omega}}_k \;\; , k \in 1 \ldots m \qquad (66)
\end{aligned}
$$

From this formulation, we see that the problem of computing the pseudo-inverse of a matrix $J_\omega$ is reduced to computing $m$ pseudo-inverses of much smaller constant-sized matrices $\hat{J}_{\omega k}$. Note that if $\dot{\mathbf{q}}$ computed in Equation (66) exactly satisfies the linear system $\boldsymbol{\omega} = J_\omega \dot{\mathbf{q}}$, it also satisfies the linear velocity relation $\mathbf{v} = J_v \dot{\mathbf{q}}$.

For systems that include translational DOFs as well, we can separately solve for the rotational DOFs as in Equation (66) and solve for the translational DOFs for any link $k$ as $\dot{\mathbf{q}}_k = J_{vk}^{+} \mathbf{v}_k$. In most of the cases, only the root joint has translational DOFs making the computation of generalized translational velocities extremely simple as $J_{v1}$ becomes an identity matrix.

## 6.2 Force conversion

The relation between the Cartesian force and the generalized force can be found in Equation (3):

$$\mathbf{Q} = \sum_k {J'_{vk}}^T \mathbf{f}_k + J_{\omega k}^T \boldsymbol{\tau}'_k = \begin{pmatrix} {J'_v}^T & J_\omega^T \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ \boldsymbol{\tau}' \end{pmatrix}, \quad \text{where } J'_{vk} = \frac{\partial \mathbf{r}_k}{\partial \mathbf{q}} \tag{67}$$

where $\mathbf{r}_k$ is the point of application of the Cartesian force $\mathbf{f}_k$ and $\boldsymbol{\tau}'_k$ is the *body torque* applied to link $k$ expressed in the world frame.

**Note.** Body torque $\boldsymbol{\tau}'_k$ is different from body torque $\boldsymbol{\tau}_k$ in Equation (48):

$$\mathbf{Q} = J_v^T \mathbf{f} + J_\omega^T \boldsymbol{\tau}$$

Here, $\boldsymbol{\tau}'_k$ is the torque applied on link $k$ in the world frame and does *not* include the torque induced by the linear forces $\mathbf{f}_k$. However, $\boldsymbol{\tau}_k$ in Equation (48) *includes* the torque $[\mathbf{r}_k - \mathbf{x}_k]\mathbf{f}_k$ due to each force $\mathbf{f}_k$ ($\mathbf{x}_k$ is the COM of the link $k$). As a result, the linear Jacobian $J_v$ in Equation (48) is computed at the COM of the respective rigid link and $J'_v$ in Equation (67) is defined for the point of application of the force. It is easy to verify that ${J'_{vk}}^T \mathbf{f}_k = J_{vk}^T \mathbf{f}_k + J_{\omega k}^T [\mathbf{r}_k - \mathbf{x}_k]\mathbf{f}_k$. i.e. $\boldsymbol{\tau}_k = \boldsymbol{\tau}'_k + [\mathbf{r}_k - \mathbf{x}_k]\mathbf{f}_k$.

Often many controllers (such as a tracking controller) find it convenient to compute the Cartesian-space *joint torques* in the local frame of the parent link rather than body torques in the world frame. Joint torque $\hat{\boldsymbol{\tau}}_k$ in the frame of a parent link $p(k)$ is defined such that positive torque in the world frame $R^0_{p(k)} \hat{\boldsymbol{\tau}}_k$ is applied to the link $k$ and negative torque $-R^0_{p(k)} \hat{\boldsymbol{\tau}}_k$ is applied to the parent link $p(k)$. Therefore, the body torque $\boldsymbol{\tau}'_k$ applied to the link $k$ can be written in terms of the joint torques as $\boldsymbol{\tau}'_k = R^0_{p(k)} \hat{\boldsymbol{\tau}}_k - \sum_l R^0_k \hat{\boldsymbol{\tau}}_l$, $\forall l : k = p(l)$. Collecting the body torques for all the rigid links in the vector $\boldsymbol{\tau}$, the relation between the body torques and the joint torques can be defined as:

$$\boldsymbol{\tau}' = D^T R \hat{\boldsymbol{\tau}} = (R^T D)^T \hat{\boldsymbol{\tau}} \tag{68}$$

where $R, D$ are defined in Equation (65). We now substitute Equation (68) in Equation (67) and get:

$$\begin{aligned}
\mathbf{Q} &= \begin{pmatrix} J_v^{'T} & J_\omega^T \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ (R^T D)^T \hat{\boldsymbol{\tau}} \end{pmatrix} \\
&= \begin{pmatrix} J_v^{'T} & (R^T D J_\omega)^T \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ \hat{\boldsymbol{\tau}} \end{pmatrix} \\
&= \begin{pmatrix} J_v^{'T} & \hat{J}_\omega^T \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ \hat{\boldsymbol{\tau}} \end{pmatrix} \quad \text{(Using Equation (65))}
\end{aligned} \tag{69}$$

Equation (69) gives the relation to convert the given cartesian forces $\mathbf{f}$ and joint torques $\hat{\boldsymbol{\tau}}$ to the generalized forces $\mathbf{Q}$.

We now describe the process to convert the given generalized forces $\mathbf{Q}$ to cartesian forces and torques. In general, the transposed Jacobian in Equation (67) can be inverted using a pseudo-inverse to get the Cartesian forces and torques. Note that the relation represents an under-constrained system when solving for $\mathbf{f}$ and $\boldsymbol{\tau}^{'}$. This is because the size of the unknowns is $6m$ and the number of constraints are $n$ with $n \leq 6m$. Therefore, we get particular solutions for the Cartesian forces and torques out of many possible solutions.

Based on the information about the form of $\mathbf{Q}$, we can solve for the Cartesian forces and torques in different ways. We describe the solutions to the following cases:

1. **General case.** In the most general case, the points of application of the Cartesian forces are not known. Therefore, we cannot use the Jacobian $J_v^{'}$ in Equation (67). This forces us to assume the points of application to be the COM of each link and compute the torques $\boldsymbol{\tau}$ instead of body torques $\boldsymbol{\tau}^{'}$. i.e. , we can invert the transposed Jacobian by computing its pseudo-inverse that results in a particular least-squares solution for $\mathbf{f}$ and $\boldsymbol{\tau}$ as:

$$\begin{pmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{pmatrix} = \begin{pmatrix} J_v{}^T & J_\omega^T \end{pmatrix}^+ \mathbf{Q} \tag{70}$$

   If the points of the force application are known, the Jacobian in Equation (69) can be inverted to obtain the forces $\mathbf{f}$ and the joint torques $\hat{\boldsymbol{\tau}}$.

2. **No linear forces.** The more common case for many controllers involves the conversion of only the joint torques from generalized to the Cartesian coordinates. Therefore, the linear forces $\mathbf{f}$ are zero and Equation (69) can be simplified further to result in the following conversion relation:

$$\begin{aligned}
\hat{\boldsymbol{\tau}} &= (\hat{J}_\omega^T)^+ \mathbf{Q} \\
\text{or } \hat{\boldsymbol{\tau}}_k &= (\hat{J}_{\omega k}^T)^+ \mathbf{Q}_k \;\; \forall k \in 1 \ldots m
\end{aligned} \tag{71}$$

   where $\mathbf{Q}_k$ denotes the components of the generalized forces corresponding to the rotational DOFs $\mathbf{q}_k$. Note that the size of the matrix $\hat{J}_{\omega k}^T$ is $n(k) \times 3$ and $n(k) \leq 3$. This implies that we get a particular least-squares solution for each $\hat{\boldsymbol{\tau}}_k$ out of possibly many solutions that would give rise to the same $\mathbf{Q}_k$ using the relation $\mathbf{Q}_k = \hat{J}_{\omega k}^T \hat{\boldsymbol{\tau}}_k$.

# 7 Recursive Inverse Dynamics

As Featherstone pointed out, inverse dynamics can be computed efficiently and much faster by exploiting the recursive structure of an articulated rigid body system. A recursive algorithm allows computation of inverse dynamics in linear time proportional to the number of links in the articulated system. In this chapter, we will use our formulation to construct a recursive inverse dynamics algorithm.

## 7.1 Dynamics in the local frame

For each body link $k$, we define $\mathbf{c}_k$ as the center of mass in the local frame and $\mathbf{d}_{c(k)[i]}$ as the vector between the joint connecting to the parent link and the joint connecting to the $i$-th child of link $k$, where $c(k)$ returns the indices of child links of the link $k$. As a shorthand, we define $\tilde{i} = c(k)[i]$ and use the notation $\mathbf{d}_{\tilde{i}}$ hereafter. Figure 2 illustrates the notations using the same structure from the previous example.

The goal of the inverse dynamics algorithm is to compute the force and the torque transmitted between links. For each link $k$, we define the force and torque received from the parent link $\mathbf{f}_k$ and $\boldsymbol{\tau}_k$. Similarly, the force and the torque from the $i$-th child are denoted as $-\mathbf{f}_{\tilde{i}}$ and $-\boldsymbol{\tau}_{\tilde{i}}$. Please see Figure 2 for illustration.

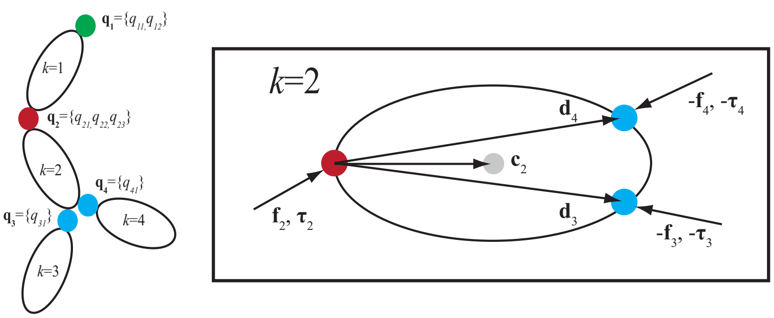


Figure 2: An articulated system.

To compute these forces and torques, let us first write down Newton-Euler equations for link $k$ in its local frame. We will use the notation $\mathbf{a}_k^\ell$ to denote a vector $\mathbf{a}$ expressed in the local frame of link $k$.

$$m_k(\dot{\mathbf{v}}_k)^\ell = \mathbf{f}_k^\ell - \sum_{\tilde{i} \in c(k)} \mathbf{R}_{\tilde{i}} \mathbf{f}_{\tilde{i}}^\ell \tag{72}$$

$$\mathbf{I}_{ck}(\dot{\boldsymbol{\omega}}_k)^\ell + \boldsymbol{\omega}_k^\ell \times \mathbf{I}_{ck}\boldsymbol{\omega}_k^\ell = \boldsymbol{\tau}_k^\ell - \mathbf{c}_k \times \mathbf{f}_k^\ell - \sum_{\tilde{i} \in c(k)} (\mathbf{R}_{\tilde{i}}\boldsymbol{\tau}_{\tilde{i}}^\ell + (\mathbf{d}_{\tilde{i}} - \mathbf{c}_k) \times (\mathbf{R}_{\tilde{i}}\mathbf{f}_{\tilde{i}}^\ell)) \tag{73}$$

The inverse dynamics algorithm visits each link twice in two recursive passes. In the first pass, the velocity and acceleration of each link is computed and expressed in the local frame. In the second pass, these terms are plugged into the above Newton-Euler equations to compute forces and torques transmitted between the links.

## 7.2 Pass 1: Compute velocity and acceleration

The articulated rigid body system can be represented as a tree structure, where every link from the root to the leaves will be visited once in Pass 1. The algorithm is recursive because the computation at each link depends on the computation of its parent link. Let us first discuss the computation for a general link and take care of the special case of the root later in this section.

Assuming the velocity and the acceleration of the parent link, $\mathbf{v}^\ell_{p(k)}$, $\boldsymbol{\omega}^\ell_{p(k)}$, $(\dot{\mathbf{v}}_{p(k)})^\ell$, $(\dot{\boldsymbol{\omega}}_{p(k)})^\ell$ , are already computed from the previous iteration, the COM of the link $k$ in the world frame is $\mathbf{W}^0_k\mathbf{c}_k$. The linear velocity of the link $k$ in Homogeneous coordinates is then expressed as:

$$\begin{aligned}\mathbf{v}_k &= \dot{\mathbf{W}}^0_k\mathbf{c}_k = \dot{\mathbf{W}}^0_{p(k)}\mathbf{W}_k\mathbf{c}_k + \mathbf{W}^0_{p(k)}\dot{\mathbf{W}}_k\mathbf{c}_k \\ &= \dot{\mathbf{W}}^0_{p(k)}(\mathbf{c}_{p(k)} + \mathbf{W}_k\mathbf{c}_k - \mathbf{c}_{p(k)}) + \mathbf{W}^0_{p(k)}\dot{\mathbf{W}}_k\mathbf{c}_k \\ &= \mathbf{v}_{p(k)} + \dot{\mathbf{W}}^0_{p(k)}(\mathbf{W}_k\mathbf{c}_k - \mathbf{c}_{p(k)}) + \mathbf{W}^0_{p(k)}\dot{\mathbf{W}}_k\mathbf{c}_k \end{aligned} \tag{74}$$

where the fourth element of the vector $\mathbf{W}_k\mathbf{c}_k - \mathbf{c}_{p(k)}$ and that of the vector $\dot{\mathbf{W}}_k\mathbf{c}_k$ are both zero. This will result in elimination of the translation part of the transformation. We can therefore express $\mathbf{v}_k$ in Cartesian space as:

$$\begin{aligned}\mathbf{v}_k &= \mathbf{v}_{p(k)} + \dot{\mathbf{R}}^0_{p(k)}(\mathbf{R}_k\mathbf{c}_k + \mathbf{d}_k - \mathbf{c}_{p(k)}) + \mathbf{R}^0_{p(k)}\dot{\mathbf{R}}_k\mathbf{c}_k \\ &= \mathbf{v}_{p(k)} + [\boldsymbol{\omega}_{p(k)}]\mathbf{R}^0_{p(k)}(\mathbf{R}_k\mathbf{c}_k + \mathbf{d}_k - \mathbf{c}_{p(k)}) + \mathbf{R}^0_{p(k)}[\hat{\boldsymbol{\omega}}_k]\mathbf{R}_k\mathbf{c}_k \quad (\text{Using } [\boldsymbol{\omega}] = \dot{\mathbf{R}}\mathbf{R}^T)\end{aligned} \tag{75}$$

So far $\mathbf{v}_k$ is computed in the world frame and we would like to transform it to the local frame.

$$\begin{aligned}\mathbf{v}^\ell_k &= \mathbf{R}^{0^T}_k\mathbf{v}_k = \mathbf{R}^T_k{\mathbf{R}^0_{p(k)}}^T\mathbf{v}_k \\ &= \mathbf{R}^T_k(\mathbf{v}^\ell_{p(k)} + [\boldsymbol{\omega}^\ell_{p(k)}](\mathbf{R}_k\mathbf{c}_k + \mathbf{d}_k - \mathbf{c}_{p(k)}) + [\hat{\boldsymbol{\omega}}_k]\mathbf{R}_k\mathbf{c}_k) \quad (\text{Using } \mathbf{R}[\boldsymbol{\omega}]\mathbf{R}^T = [\mathbf{R}\boldsymbol{\omega}]) \\ &= \mathbf{R}^T_k(\mathbf{v}^\ell_{p(k)} + [\boldsymbol{\omega}^\ell_{p(k)}](\mathbf{d}_k - \mathbf{c}_{p(k)})) + \mathbf{R}^T_k(([\boldsymbol{\omega}^\ell_{p(k)}] + [\hat{\boldsymbol{\omega}}_k])\mathbf{R}_k\mathbf{c}_k) \\ &= \mathbf{R}^T_k(\mathbf{v}^\ell_{p(k)} + \boldsymbol{\omega}^\ell_{p(k)} \times (\mathbf{d}_k - \mathbf{c}_{p(k)})) + \boldsymbol{\omega}^\ell_k \times \mathbf{c}_k \quad (\text{Using } \mathbf{R}[\boldsymbol{\omega}]\mathbf{R}^T = [\mathbf{R}\boldsymbol{\omega}])\end{aligned} \tag{76}$$

Similarly, we can compute the angular velocity of the link $k$ in the world frame and then transform it to the local frame.

$$\begin{aligned}[\boldsymbol{\omega}_k] &= \dot{\mathbf{R}}^0_k\mathbf{R}^{0^T}_k = (\mathbf{R}^0_{p(k)}\dot{\phantom{x}}\mathbf{R}_k)(\mathbf{R}^T_k{\mathbf{R}^0_{p(k)}}^T) = \dot{\mathbf{R}}^0_{p(k)}\mathbf{R}_k\mathbf{R}^T_k{\mathbf{R}^0_{p(k)}}^T + \mathbf{R}^0_{p(k)}\dot{\mathbf{R}}_k\mathbf{R}^T_k{\mathbf{R}^0_{p(k)}}^T \\ &= [\boldsymbol{\omega}_{p(k)}] + \mathbf{R}^0_{p(k)}[\hat{\boldsymbol{\omega}}_k]{\mathbf{R}^0_{p(k)}}^T\end{aligned} \tag{77}$$

$$\begin{aligned}[\boldsymbol{\omega}^\ell_k] &= [\mathbf{R}^{0^T}_k\boldsymbol{\omega}_k] = \mathbf{R}^{0^T}_k[\boldsymbol{\omega}_k]\mathbf{R}^0_k \\ &= \mathbf{R}^T_k({\mathbf{R}^0_{p(k)}}^T[\boldsymbol{\omega}_{p(k)}]\mathbf{R}^0_{p(k)} + [\hat{\boldsymbol{\omega}}_k])\mathbf{R}_k = \mathbf{R}^T_k([\boldsymbol{\omega}^\ell_{p(k)}] + [\hat{\boldsymbol{\omega}}_k])\mathbf{R}_k \\ \boldsymbol{\omega}^\ell_k &= \mathbf{R}^T_k(\boldsymbol{\omega}^\ell_{p(k)} + \hat{\boldsymbol{\omega}}_k)\end{aligned} \tag{78}$$

Next, we compute the linear and angular acceleration for the link $k$. Note that the linear acceleration must be computed in the world frame first and then transformed into the local frame. If we instead take the time derivative on $\mathbf{v}_k^\ell$, i.e. $\dot{\mathbf{v}}_k^\ell$, the result is different from the true linear acceleration $(\dot{\mathbf{v}}_k)^\ell$, as the former does not take into account the Coriolis forces due to the moving frame.

$$
\begin{aligned}
\mathbf{v}_k &= \mathbf{R}_k^0\mathbf{v}_k^\ell = \mathbf{R}_{p(k)}^0(\mathbf{v}_{p(k)}^\ell + \boldsymbol{\omega}_{p(k)}^\ell \times (\mathbf{d}_k - \mathbf{c}_{p(k)})) + \mathbf{R}_k^0\boldsymbol{\omega}_k^\ell \times \mathbf{c}_k \\
(\dot{\mathbf{v}}_k)^\ell &= \mathbf{R}_k^{0T}\dot{\mathbf{R}}_{p(k)}^0(\mathbf{v}_{p(k)}^\ell + \boldsymbol{\omega}_{p(k)}^\ell \times (\mathbf{d}_k - \mathbf{c}_{p(k)})) + \mathbf{R}_k^T(\dot{\mathbf{v}}_{p(k)}^\ell + \dot{\boldsymbol{\omega}}_{p(k)}^\ell \times (\mathbf{d}_k - \mathbf{c}_{p(k)})) \\
&\quad + \mathbf{R}_k^{0T}\dot{\mathbf{R}}_k^0(\boldsymbol{\omega}_k^\ell \times \mathbf{c}_k) + \dot{\boldsymbol{\omega}}_k^\ell \times \mathbf{c}_k \\
&= \mathbf{R}_k^T(\dot{\mathbf{v}}_{p(k)}^\ell + \boldsymbol{\omega}_{p(k)}^\ell \times \mathbf{v}_{p(k)}^\ell + \dot{\boldsymbol{\omega}}_{p(k)}^\ell \times (\mathbf{d}_k - \mathbf{c}_{p(k)}) + \boldsymbol{\omega}_{p(k)}^\ell \times (\boldsymbol{\omega}_{p(k)}^\ell \times (\mathbf{d}_k - \mathbf{c}_{p(k)}))) \\
&\quad + \boldsymbol{\omega}_k^\ell \times (\boldsymbol{\omega}_k^\ell \times \mathbf{c}_k) + \dot{\boldsymbol{\omega}}_k^\ell \times \mathbf{c}_k \quad (\text{Using } \mathbf{R}_k^{0T}\dot{\mathbf{R}}_k^0 = [\boldsymbol{\omega}_k^\ell]]) \\
&= \mathbf{R}_k^T((\dot{\mathbf{v}}_{p(k)})^\ell + \dot{\boldsymbol{\omega}}_{p(k)}^\ell \times (\mathbf{d}_k - \mathbf{c}_{p(k)}) + \boldsymbol{\omega}_{p(k)}^\ell \times (\boldsymbol{\omega}_{p(k)}^\ell \times (\mathbf{d}_k - \mathbf{c}_{p(k)}))) \\
&\quad + \boldsymbol{\omega}_k^\ell \times (\boldsymbol{\omega}_k^\ell \times \mathbf{c}_k) + \dot{\boldsymbol{\omega}}_k^\ell \times \mathbf{c}_k \quad (\text{Using } (\dot{\mathbf{v}}_{p(k)})^\ell = \dot{\mathbf{v}}_{p(k)}^\ell + \boldsymbol{\omega}_{p(k)}^\ell \times \mathbf{v}_{p(k)}^\ell)
\end{aligned} \tag{79}
$$

In contrast, the angular acceleration $(\dot{\boldsymbol{\omega}}_k)^\ell$ is the same as $\dot{\boldsymbol{\omega}}_k^\ell$.

$$
(\dot{\boldsymbol{\omega}}_k)^\ell = \mathbf{R}_k^T((\dot{\boldsymbol{\omega}}_{p(k)})^\ell + \dot{\hat{\boldsymbol{\omega}}}_k + \boldsymbol{\omega}_{p(k)}^\ell \times \hat{\boldsymbol{\omega}}_k) \tag{80}
$$

**Base case:** The base case of this recursive pass computes the velocity and the acceleration of the root link. We can simplified the computation as follows:

$$
\begin{aligned}
\mathbf{v}_0^\ell &= \boldsymbol{\omega}_0^\ell \times \mathbf{c}_0 \\
\boldsymbol{\omega}_0^\ell &= \mathbf{R}_0^T\hat{\boldsymbol{\omega}}_0 \\
(\dot{\mathbf{v}}_0)^\ell &= \boldsymbol{\omega}_0^\ell \times (\boldsymbol{\omega}_0^\ell \times \mathbf{c}_0) + (\dot{\boldsymbol{\omega}}_0)^\ell \times \mathbf{c}_0 \\
(\dot{\boldsymbol{\omega}}_0)^\ell &= \mathbf{R}_0^T\dot{\hat{\boldsymbol{\omega}}}_0
\end{aligned} \tag{81}
$$

**Translational degrees of freedom:** So far we assume that the links are connected by only rotational DOFs, but the formulation can be easily generalized to translational DOFs. Let us rewrite Equation 74 with the assumption that $\mathbf{W}_k$ has only translational DOFs $\mathbf{q}_k$, i.e. $\mathbf{R}_k = \mathbf{I}_3$ and $\mathbf{d}_k = \mathbf{q}_k$.

$$
\mathbf{v}_k = \mathbf{v}_{p(k)} + \dot{\mathbf{R}}_{p(k)}^0(\mathbf{c}_k + \mathbf{q}_k - \mathbf{c}_{p(k)}) + \mathbf{R}_{p(k)}^0\dot{\mathbf{q}}_k \tag{82}
$$

Transforming into the local frame of the link $k$, we get

$$
\mathbf{v}_k^\ell = \mathbf{v}_{p(k)}^\ell + \boldsymbol{\omega}_{p(k)}^\ell \times (\mathbf{c}_k + \mathbf{q}_k - \mathbf{c}_{p(k)}) + \dot{\mathbf{q}}_k \tag{83}
$$

The angular velocity and accelerations can be derived in a similar way.

$$
\begin{aligned}
\boldsymbol{\omega}_k^\ell &= \boldsymbol{\omega}_{p(k)}^\ell \\
\dot{\mathbf{v}}_k^\ell &= \dot{\mathbf{v}}_{p(k)}^\ell + \dot{\boldsymbol{\omega}}_{p(k)}^\ell \times (\mathbf{c}_k + \mathbf{q}_k - \mathbf{c}_{p(k)}) + \boldsymbol{\omega}_{p(k)}^\ell \times \dot{\mathbf{q}}_k + \ddot{\mathbf{q}}_k \\
\dot{\boldsymbol{\omega}}_k^\ell &= \dot{\boldsymbol{\omega}}_{p(k)}^\ell
\end{aligned} \tag{84}
$$

### 7.3 Pass 2: Compute force and torque

The second pass computes forces and torques transmitted between body links. The algorithm visits each link once from the leaf nodes to the root. At each iteration, we compute $\mathbf{f}_k^\ell$ and $\boldsymbol{\tau}_k^\ell$, given the velocity and the acceleration computed by Pass 1 and all the forces and torques from the child links. Specifically, we will plug $\mathbf{v}_k^\ell$, $\boldsymbol{\omega}_k^\ell$, $(\dot{\mathbf{v}}_k)^\ell$, $(\dot{\boldsymbol{\omega}}_k)^\ell$, and all the $\mathbf{f}_{\tilde{i}}^\ell$ and $\boldsymbol{\tau}_{\tilde{i}}^\ell$ into Equation 72 and Equation 73.

**Base case:** The leaf nodes have no child links connected to them. Therefore, $\mathbf{f}_{\tilde{i}}$ and $\boldsymbol{\tau}_{\tilde{i}}$ are zero at each leaf node. Similarly for the root link, $\mathbf{f}_0$ and $\boldsymbol{\tau}_0$ are zero.

**Gravity:** Instead of treating gravity as an external force, we can consider the effect of gravity by conveniently offsetting the linear acceleration of the root link by $-\mathbf{g}$.

$$(\dot{\mathbf{v}}_0)^\ell = \boldsymbol{\omega}_0^\ell \times (\boldsymbol{\omega}_0^\ell \times \mathbf{c}_0) + (\dot{\boldsymbol{\omega}}_0)^\ell \times \mathbf{c}_0 - \mathbf{R}_0^T \mathbf{g} \tag{85}$$

This is equivalent to adding a fictitious force, $-m_k\mathbf{g}$, to each link. The rest of the algorithm remains unchanged.

## Acknowledgements

We thank Jeff Bingham, Stelian Coros, Marco da Silva, and Yuting Ye for proofreading this document and providing valuable comments.