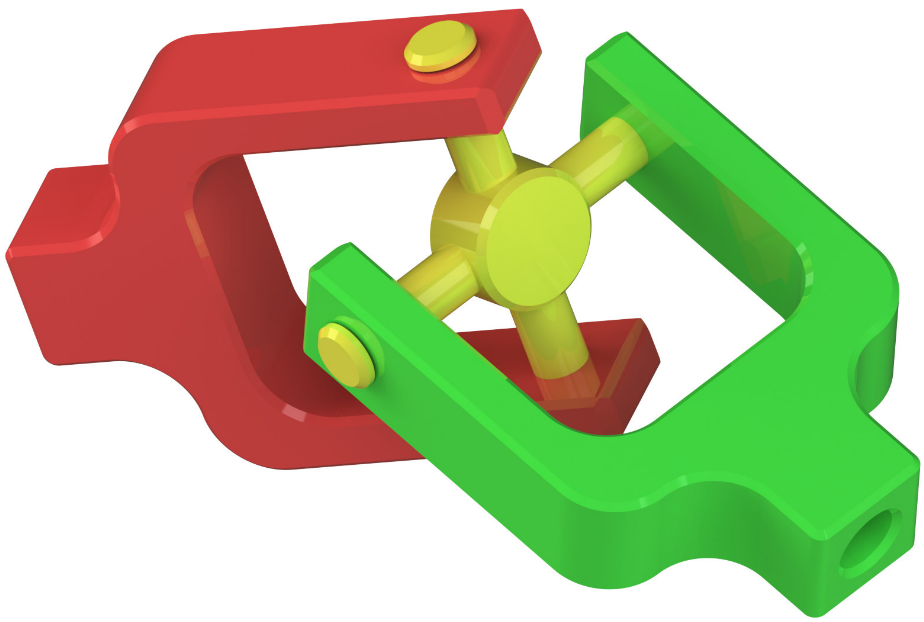
3D Constraint Derivations for Impulse Solvers

Marijn Tamis

July 1, 2015 v1.00



Abstract

It is common to solve constrained rigid body systems using velocity constraints. However it is natural to express constraints as a relation of positions, making it necessary to take their derivative before solving them. Many solvers also expect the constraint equations in a form where velocities can be factored out. This document is a reference for the derivation of such constraints.

Contents

1 Introduction

This document shows how to setup position constraints for a number of different joint types, and shows how to derive their velocity constraints. We specifically aim to describe the constraint equations so they can be directly implemented in impulse based constraint solvers like described in [5]. However, most of the equations are general and can be used in other solvers as well with little or no modification. In this document we use λ as the signed magnitude of the constraint impulse (not force) unless stated otherwise.

It is assumed that the reader is already familiar with constraint solvers like described in [5] and [3], as this document will not go into detail.

1.1 Notational Conventions

Here we define the notational conventions used throughout this document.

Element	Typesetting	Examples
Scalars	lowercase italics	m, ω, Ω
Vectors	lowercase fat upright	$\mathbf{x}, \boldsymbol{\theta}$
Quaternions	lowercase fat upright	$\mathbf{o}, \mathbf{q}$
Matrices	uppercase fat upright	$\mathbf{I}, \mathbf{R}$
Units	upright	$\mathrm{kg\,m\,s^{-1}}$
Functions	upright	$\mathrm{c}(\ldots)$

1.2 Common Variable Definitions

Symbol	Type	Unit	Description
m_a	Scalar	kg	Mass of body a
$\mathbf{M}$	6n x 6n matrix	kg and $\mathrm{kg\,m^2}$	System mass containing m and $\mathbf{I}$ for all n bodies affected by the constraint.
$\mathbf{I}_a$	3x3 matrix	$\mathrm{kg\,m^2}$	Inertia tensor of body a
$\mathbf{x}_a$	$\mathbb{R}^3$ vector	m	Position of the center of mass of body a
$\mathbf{q}_a$	Quaternion	rad	Orientation of body a
$\mathbf{p}_a$	$\mathbb{R}^3$ vector	m	Point to which the constraint is attached to body a in world space
$\mathbf{p}$	$\mathbb{R}^{6n}$ vector	$\mathrm{kg\,m\,s^{-1}}$ and $\mathrm{kg\,m^2\,s^{-1}}$	Constraint impulse containing linear and angular impulses for the n attached bodies
$\mathbf{f}$	$\mathbb{R}^3$ vector	$\mathrm{kg\,m\,s^{-2}}$ or N	Force
$\mathbf{v}_a$	$\mathbb{R}^3$ vector	$\mathrm{m\,s^{-1}}$	Linear velocity of body a
$\mathbf{v}$	$\mathbb{R}^{6n}$ vector	$\mathrm{m\,s^{-1}}$ and $\mathrm{rad\,s^{-1}}$	System velocity, containing $\mathbf{v}$ and $\boldsymbol{\omega}$ for all bodies affected by the constraint
$\boldsymbol{\omega}_a$	$\mathbb{R}^3$ vector	$\mathrm{rad\,s^{-1}}$	Angular velocity of body a
t	Scalar	s	(Current) Time
Δt	Scalar	s	Time step
$\mathrm{c}(\ldots)$	Function	N/A	Position constraint
$\dot{\mathrm{c}}(\ldots)$	Function	$\mathrm{m\,s^{-1}}$	Velocity constraint
$\mathbf{j}_c$	$\mathbb{R}^{12}$ vector	None and m	A single row of $\mathbf{J}$
$\mathbf{J}$	$d \times 12$ Matrix	None and m	All Jacobians for the constraint. Also written as $\mathbf{j}$ when $\mathbf{J}$ contains only one row.
$\mathbf{r}_a$	$\mathbb{R}^3$ vector	m	Offset between point i and center of mass of body a. $\mathbf{r}_a = \mathbf{p}_a - \mathbf{x}_a$
$[\mathbf{a}]_\times$	3×3 matrix	N/A	Cross product matrix of $\mathbf{a}$ so that $[\mathbf{a}]_\times \mathbf{b} = \mathbf{a} \times \mathbf{b}$
$\mathbf{D}$	matrix	N/A	Identity matrix, usually 3×3

2 Impulses

Here we show how a sequential impulse solver calculates the constraint reaction impulses. Additional derivations can be found in Appendix B.

2.1 Derivation of Impulse Solution

In this section we show the derivation of the constraint reaction impulse in detail. Specifically we show how to get from the formulas of slide 54 of [6] to the impulse solution on slide 55.

We start with the following system of equations:

$$\mathbf{v} = \overline{\mathbf{v}} + \mathbf{M}^{-1}\mathbf{p}_c \tag{2.1}$$

$$\mathbf{p}_c = \mathbf{J}^{\mathsf{T}}\lambda \tag{2.2}$$

$$0 = \mathbf{J}\mathbf{v} + b \tag{2.3}$$

Where $\overline{\mathbf{v}}$ contains the pre-collision velocities of body the bodies, $\mathbf{v}$ contains the post-collision velocities, $\mathbf{p}_c$ is the constraint impulse, $\mathbf{J}$ is the constraint Jacobian, and $\mathbf{M}^{-1}$ is the inverse mass matrix. (2.1) is an Euler step where the new velocity is obtained by integrating the constraint impulse, (2.2) is the equation of virtual work, and (2.3) is the velocity constraint. b is a velocity bias and can be used for various purposes (eg. Baumgarte stabilization and restitution).

First we substitute (2.3) with (2.1):

$$\begin{aligned} 0 &= \mathbf{J}\left(\overline{\mathbf{v}} + \mathbf{M}^{-1}\mathbf{p}_c\right) + b \\ 0 &= \mathbf{J}\overline{\mathbf{v}} + \mathbf{J}\mathbf{M}^{-1}\mathbf{p}_c + b \\ -\mathbf{J}\mathbf{M}^{-1}\mathbf{p}_c &= \mathbf{J}\overline{\mathbf{v}} + b \end{aligned}$$

Now we substitute Equation (2.2):

$$-\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\mathsf{T}}\lambda = \mathbf{J}\overline{\mathbf{v}} + b$$

And finally we divide by $-\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\mathsf{T}}$ to solve for λ:

$$\lambda = \frac{\mathbf{J}\overline{\mathbf{v}} + b}{-\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\mathsf{T}}} \tag{2.4}$$

(2.2) can be used to calculate the impulse from λ.

2.2 Calculating the Effective Mass

Here we show how to calculate the effective mass matrix for a general Jacobian so we do not have to repeat these steps for every constraint.

We use the following general Jacobian for constraints between two bodies, a and b:

$$\mathbf{J} = \begin{bmatrix} -\mathbf{L}_a, -\mathbf{A}_a, \mathbf{L}_b, \mathbf{A}_b \end{bmatrix}$$

Where $\mathbf{L}_i$ is the linear component of the Jacobian and $\mathbf{A}_i$ is the angular component. $\mathbf{L}_i$ and $\mathbf{A}_i$ are matrices or row vectors, depending on the constraint.

We give $\mathbf{M}^{-1}$ for reference:

$$\mathbf{M}^{-1} = \begin{bmatrix} m_a^{-1}\mathbf{D} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I_a}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & m_b^{-1}\mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I_b}^{-1} \end{bmatrix} \tag{2.5}$$

Expanding the products gives us the effective mass:

$$\begin{aligned} \mathbf{J}\mathbf{M}^{-1} &= \begin{bmatrix} -\mathbf{L}_a m_a^{-1}, & -\mathbf{A}_a \mathbf{I}_a^{-1}, & \mathbf{L}_b m_b^{-1}, & \mathbf{A}_b \mathbf{I}_a^{-1} \end{bmatrix} \\ \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^\intercal &= \mathbf{L}_a \mathbf{L}_a^\intercal m_a^{-1} + \mathbf{A}_a \mathbf{I}_a^{-1} \mathbf{A}_a^\intercal + \mathbf{L}_b \mathbf{L}_b^\intercal m_b^{-1} + \mathbf{A}_b \mathbf{I}_a^{-1} \mathbf{A}_b^\intercal \end{aligned} \tag{2.6}$$

Note that the signs of $\mathbf{A}_i$ and $\mathbf{L}_i$ cancel out, and that $\mathbf{L}_i \mathbf{L}_i^\intercal$ becomes identity if it is a unit vector or a rotation matrix.

2.3 Baumgarte Stabilization

In this section we show how to calculate the Baumgarte stabilization value b for equation 2.4 for all constraints.

Baumgarte stabilization will effectively use b to bias the velocity constraint so that the bodies will move towards a correct configuration. The position constraint determines this correct configuration, giving us:

$$b = \frac{\mathrm{c}(\dots)}{\Delta t} \beta \tag{2.7}$$

Where $\beta \in [0, 1]$ is the Baumgarte factor. Increasing β forces the constraints to be corrected faster and also causes instability or oscillation when chosen too high.

b becomes a vector if $\mathrm{c}(\dots)$ has a range of $\mathbb{R}^3$.

2.4 Soft Constraints Using Impulses

Here we show the derivation of the constraint reaction impulse for constraints with softness. We will obtain an equation that can be used instead of (2.4). This derivation based on [4], also see [7].

We begin with writing Newton's second law using impulses:

$$\mathbf{p} = \mathbf{M}\dot{\mathbf{v}}\Delta t \approx \mathbf{M}\left(\mathbf{v} - \overline{\mathbf{v}}\right) = \mathbf{J}^\mathsf{T}\boldsymbol{\lambda} \tag{2.8}$$

Where $\overline{\mathbf{v}}$ is the pre-collision velocity and $\mathbf{p}$ and $\boldsymbol{\lambda}$ are impulses.

We modify the velocity constraint to include both softness and Baumgarte stabilization:

$$\dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = \mathbf{J}\mathbf{v} + \boldsymbol{\gamma} \cdot \frac{\boldsymbol{\lambda}}{\Delta t} + \frac{\boldsymbol{\beta} \cdot \mathbf{c}}{\Delta t} = 0 \tag{2.9}$$

Where $\boldsymbol{\gamma}$ is the softness, $\mathbf{c}$ is the position constraint error.

Everything is known except for $\mathbf{v}$ and $\boldsymbol{\lambda}$. We first solve (2.8) for $\mathbf{v}$ in terms of $\boldsymbol{\lambda}$:

$$\begin{aligned} \mathbf{v} - \overline{\mathbf{v}} &= \mathbf{M}^{-1}\mathbf{J}^\mathsf{T}\boldsymbol{\lambda} \\ \mathbf{v} &= \mathbf{M}^{-1}\mathbf{J}^\mathsf{T}\boldsymbol{\lambda} + \overline{\mathbf{v}} \end{aligned} \tag{2.10}$$

Then we substitute this in to (2.9) and we collect the coefficients of lambda:

$$\begin{aligned} \dot{\mathrm{c}}(\ldots) &= \mathbf{J}\left(\mathbf{M}^{-1}\mathbf{J}^\mathsf{T}\boldsymbol{\lambda} + \overline{\mathbf{v}}\right) + \boldsymbol{\gamma} \cdot \frac{\boldsymbol{\lambda}}{\Delta t} + \frac{\boldsymbol{\beta} \cdot \mathbf{c}}{\Delta t} \\ &= \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^\mathsf{T}\boldsymbol{\lambda} + \mathbf{J}\overline{\mathbf{v}} + \frac{\gamma}{\Delta t} \cdot \boldsymbol{\lambda} + \frac{\boldsymbol{\beta} \cdot \mathbf{c}}{\Delta t} \\ &= \left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^\mathsf{T} + \frac{\gamma}{\Delta t}\right) \cdot \boldsymbol{\lambda} + \mathbf{J}\overline{\mathbf{v}} + \frac{\boldsymbol{\beta} \cdot \mathbf{c}}{\Delta t} \end{aligned} \tag{2.11}$$

We define $\mathbf{K}$:

$$\mathbf{K} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^\mathsf{T} + \frac{\gamma}{\Delta t} \tag{2.12}$$

Which gives:

$$\mathbf{K}\boldsymbol{\lambda} = -\mathbf{J}\overline{\mathbf{v}} - \frac{\boldsymbol{\beta} \cdot \mathbf{c}}{\Delta t} \tag{2.13}$$

We calculate the delta impulse to satisfy the constraint, instead of calculating the whole impulse in each iteration. We use $\boldsymbol{\lambda}$ as the accumulated impulse and $\Delta\boldsymbol{\lambda}$ as the incremental impulse update. This gives us the following formula:

$$\mathbf{M}\left(\mathbf{v}_{\text{new}} - \mathbf{v}_{\text{damaged}}\right) = \mathbf{J}^\mathsf{T}\Delta\boldsymbol{\lambda} \tag{2.14}$$

Where $\mathbf{v}_{\text{damaged}}$ is the velocity currently calculated for our bodies which may be violating our constraint due to other constraints evaluated previously, and $\mathbf{v}_{\text{new}}$ is the velocity we want to achieve to satisfy the current constraint. Solving for $\mathbf{v}_{\text{new}}$ gives:

$$\mathbf{v}_{\text{new}} = \mathbf{M}^{-1}\mathbf{J}^\mathsf{T}\Delta\boldsymbol{\lambda} + \mathbf{v}_{\text{damaged}} \tag{2.15}$$

Our velocity constraint, now using $\Delta\boldsymbol{\lambda}$, is witter as follows:

$$\dot{\mathrm{c}}(\ldots) = \mathbf{J}\mathbf{v}_{\text{new}} + \boldsymbol{\gamma} \cdot \frac{\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}}{\Delta t} + \frac{\boldsymbol{\beta} \cdot \mathbf{c}}{\Delta t} = 0 \tag{2.16}$$

We substitute (2.15) into (2.16) and solve for $\Delta\boldsymbol{\lambda}$:

$$\begin{aligned}
\mathbf{J}\left(\mathbf{M}^{-1}\mathbf{J}^{\mathsf{T}}\Delta\boldsymbol{\lambda} + \mathbf{v}_{\text{damaged}}\right) + \boldsymbol{\gamma}\cdot\frac{\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}}{\Delta t} + \frac{\boldsymbol{\beta}\cdot\mathbf{c}}{\Delta t} &= 0 \\
\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\mathsf{T}}\Delta\boldsymbol{\lambda} + \mathbf{J}\mathbf{v}_{\text{damaged}} + \frac{\gamma}{\Delta t}\cdot\boldsymbol{\lambda} + \frac{\gamma}{\Delta t}\cdot\Delta\boldsymbol{\lambda} + \frac{\boldsymbol{\beta}\cdot\mathbf{c}}{\Delta t} &= 0 \\
\left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\mathsf{T}} + \frac{\gamma}{\Delta t}\right)\cdot\Delta\boldsymbol{\lambda} &= -\mathbf{J}\mathbf{v}_{\text{damaged}} - \frac{\gamma}{\Delta t}\cdot\boldsymbol{\lambda} - \frac{\boldsymbol{\beta}\cdot\mathbf{c}}{\Delta t} \\
\mathbf{K}\Delta\boldsymbol{\lambda} &= -\mathbf{J}\mathbf{v}_{\text{damaged}} - \frac{\gamma}{\Delta t}\cdot\boldsymbol{\lambda} - \frac{\boldsymbol{\beta}\cdot\mathbf{c}}{\Delta t} \\
\Delta\boldsymbol{\lambda} &= \frac{-\mathbf{J}\mathbf{v}_{\text{damaged}} - \frac{\gamma}{\Delta t}\cdot\boldsymbol{\lambda} - \frac{\boldsymbol{\beta}\cdot\mathbf{c}}{\Delta t}}{\mathbf{K}} \qquad (2.17)
\end{aligned}$$

2.5 Dimensional Analysis of the Soft Constraint

Here we show what the units are of the $\boldsymbol{\gamma}$ and β constants from previous section.

The unit of the velocity constraint is $\mathrm{m\,s^{-1}}$. We can derive the units of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ from this fact as follows:

$$\left[\mathbf{J}\mathbf{v} + \boldsymbol{\gamma}\cdot\frac{\boldsymbol{\lambda}}{\Delta t} + \frac{\boldsymbol{\beta}\cdot\mathbf{c}}{\Delta t}\right] = \mathrm{m\,s^{-1}}$$

This gives:

$$\begin{aligned}
\left[\gamma\cdot\frac{\boldsymbol{\lambda}}{\Delta t}\right] &= \mathrm{m\,s^{-1}} \\
[\gamma]\cdot\left[\frac{\boldsymbol{\lambda}}{\Delta t}\right] &= \mathrm{kg^{-1}\,s}\cdot\mathrm{kg\,m\,s^{-2}}
\end{aligned}$$

And

$$\begin{aligned}
\left[\frac{\boldsymbol{\beta}\cdot\mathbf{c}}{\Delta t}\right] &= \mathrm{m\,s^{-1}} \\
[\boldsymbol{\beta}]\cdot\left[\frac{\mathbf{c}}{\Delta t}\right] &= 1\cdot\mathrm{m\,s^{-1}}
\end{aligned}$$

3 Constraint Equations

In this section we derive various different constraints. We divided the constraints in the following categories:

- Section 3.1 discusses constraints for linear motion.
- Section 3.2 discusses constraints for angular motion.
- Section 3.3 discusses some constraint formulations that did not prove stable.

We have chosen for this separation to avoid repeating constraint definitions as it is often necessary to combine constraints from Section 3.1 and Section 3.2 to completely model a specific joint.

3.1 Linear Constraints

This section describes a number of linear constraints. With linear constraints we mean constraints that primarily constrain positional aspects of bodies.

3.1.1 Point distance constraint

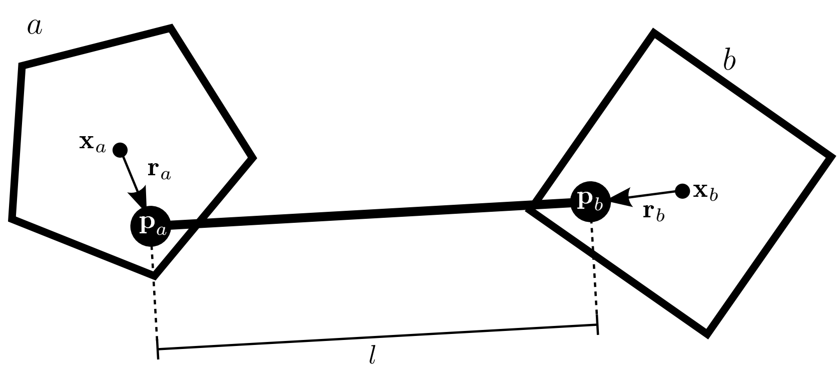


Figure 3.1: Point distance constraint.

The point distance constraint ensures that the distance between point $\mathbf{p}_a$ and $\mathbf{p}_b$ is equal to l (see figure 3.1).

This constraint will only act along one axis. This axis is calculated each frame as follows:

$$\mathbf{n} = \frac{\mathbf{p}_b - \mathbf{p}_a}{|\mathbf{p}_b - \mathbf{p}_a|} \tag{3.1}$$

Note that this will result in a division by zero when the points overlap. Pick another axis in this case.

We write the position constraint as follows:

$$\mathrm{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = (\mathbf{p}_b - \mathbf{p}_a) \cdot \mathbf{n} - l \tag{3.2}$$

The velocity constraint then becomes:

$$\begin{aligned}
\dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= (\mathbf{v}_b + \boldsymbol{\omega}_b \times \mathbf{r}_b - \mathbf{v}_a - \boldsymbol{\omega}_a \times \mathbf{r}_a) \cdot \mathbf{n} \\
&= \mathbf{v}_b \mathbf{n} + (\boldsymbol{\omega}_b \times \mathbf{r}_b)\, \mathbf{n} - \mathbf{v}_a \mathbf{n} - (\boldsymbol{\omega}_a \times \mathbf{r}_a)\, \mathbf{n} \\
&= \mathbf{v}_b \mathbf{n} + \boldsymbol{\omega}_b\, (\mathbf{r}_b \times \mathbf{n}) - \mathbf{v}_a \mathbf{n} - \boldsymbol{\omega}_a\, (\mathbf{r}_a \times \mathbf{n}) \\
\mathbf{J}\mathbf{v} &= \begin{bmatrix} -\mathbf{n} \\ -(\mathbf{r}_a \times \mathbf{n}) \\ \mathbf{n} \\ (\mathbf{r}_b \times \mathbf{n}) \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{v}_a \\ \boldsymbol{\omega}_a \\ \mathbf{v}_b \\ \boldsymbol{\omega}_b \end{bmatrix}
\end{aligned} \tag{3.3}$$

Filling in (2.6) gives the effective mass:

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\mathsf{T}} = m_a^{-1} + \left(\mathbf{I}_a^{-1} (\mathbf{r}_a \times \mathbf{n})\right) \cdot (\mathbf{r}_a \times \mathbf{n}) + m_b^{-1} + \left(\mathbf{I}_b^{-1} (\mathbf{r}_b \times \mathbf{n})\right) \cdot (\mathbf{r}_a \times \mathbf{n}) \tag{3.4}$$

3.1.2 Contact constraint

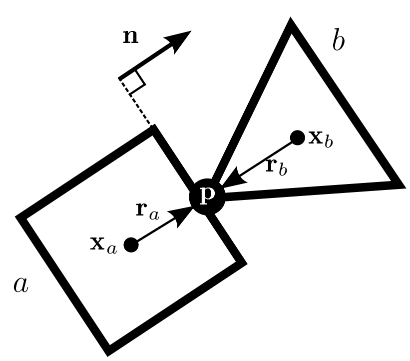


Figure 3.2: Contact constraint.

This linear constraint will force the distance between points $\mathbf{p}_a$ and $\mathbf{p}_b$ to be zero along the axis $\mathbf{n}$. This constraint is also used to model contacts by enforcing the inequality $0 < \lambda < +\infty$ and defining $\mathbf{n}$ in the direction from body a towards body b (see Figure 3.2).

We write the position constraint as follows:

$$c(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = (\mathbf{p}_b - \mathbf{p}_a) \cdot \mathbf{n} \tag{3.5}$$

Which gives us the following velocity constraint:

$$\dot{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = (\mathbf{v}_b + \boldsymbol{\omega}_b \times \mathbf{r}_b - \mathbf{v}_a - \boldsymbol{\omega}_a \times \mathbf{r}_a) \cdot \mathbf{n}$$

This leaves us with the same equation as (3.3).

3.1.3 Full position constraint

This constraint ensures that points $\mathbf{p}_a$ and $\mathbf{p}_b$ overlap. This constraint can be build from 3 linear constraints of the same form as shown in Section 3.1.2 along 3 basis vectors, or it can be written in matrix form as shown here. This matrix constraint can easily be solved as a whole, resulting in better convergence.

We define the position constraint as follows:

$$c(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = \mathbf{p}_b - \mathbf{p}_a \tag{3.6}$$

Note that the range of this function are vectors of $\mathbb{R}^3$.

This gives the following velocity constraint:

$$\begin{aligned} \dot{c}(\ldots) =& \mathbf{v}_b + \boldsymbol{\omega}_b \times \mathbf{r}_b - \mathbf{v}_a - \boldsymbol{\omega}_a \times \mathbf{r}_a \\ =& \mathbf{v}_b - [\mathbf{r}_b]_\times \boldsymbol{\omega}_b - \mathbf{v}_a + [\mathbf{r}_a]_\times \boldsymbol{\omega}_a \\ \mathbf{J}\mathbf{v} =& \begin{bmatrix} -\mathbf{D} \\ [\mathbf{r}_a]_\times^\intercal \\ \mathbf{D} \\ -[\mathbf{r}_b]_\times^\intercal \end{bmatrix}^\intercal \begin{bmatrix} \mathbf{v}_a \\ \boldsymbol{\omega}_a \\ \mathbf{v}_b \\ \boldsymbol{\omega}_b \end{bmatrix} \end{aligned} \tag{3.7}$$

Note that $[\mathbf{r}]_\times$ is a cross product matrix as shown in Section A.2.3.

Filling in (2.6) gives the effective mass matrix:

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^\intercal = \left[m_a^{-1}\mathbf{D} + [\mathbf{r}_a]_\times \mathbf{I}_a^{-1} [\mathbf{r}_a]_\times^\intercal + m_b^{-1}\mathbf{D} + [\mathbf{r}_b]_\times \mathbf{I}_b^{-1} [\mathbf{r}_b]_\times^\intercal\right] \tag{3.8}$$

3.1.4 Slider constraint

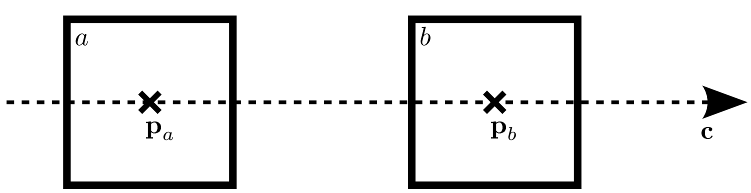


Figure 3.3: A slider constraint forcing bodies a and b to stay on axis $\mathbf{c}$.

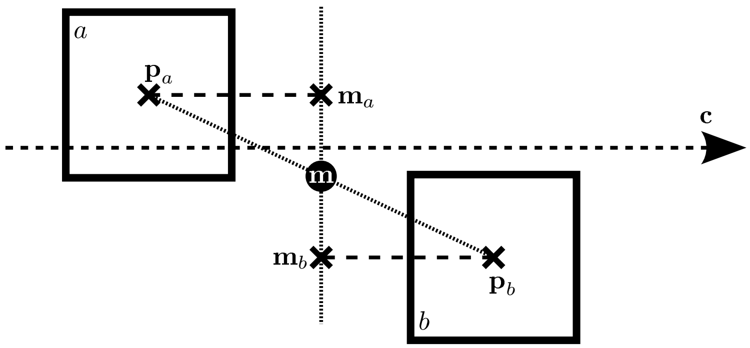


Figure 3.4: The slider constraint in a violated configuration.

The slider constraint (or prismatic joint) constraints bodies a and b to a single degree of freedom so that they can only move linearly along axis $\mathbf{c}$ (see Figure 3.3). $\mathbf{c}$ is usually calculated each frame from one of the body orientations as follows:

$$\mathbf{c} = \mathbf{R}_a \mathbf{c}_l \tag{3.9}$$

Where $\mathbf{c}_l$ is the slider axis in local space of body a and $\mathbf{R}_a$ is the local to world rotation matrix of body a.

Using the same approach as in Section 3.1.3 will yield incorrect results when the constraint is violated. This can easily be demonstrated with an example: If the constraint attachment points $\mathbf{p}_a$ and $\mathbf{p}_b$ lie on the centers of mass of their respective bodies, $\mathbf{r}_a$ and $\mathbf{r}_b$ will be $\mathbf{0}$, always resulting in $\mathbf{0}$ torque. This would cause the constraint to only be corrected with linear motion, which is incorrect.

Instead we use the middle point $\mathbf{m}$ to calculate two new attachment points $\mathbf{m}_a$ and $\mathbf{m}_a$ (see Figure 3.4) as follows:

$$\mathbf{m} = \frac{1}{2}\left(\mathbf{p}_a + \mathbf{p}_b\right) \tag{3.10}$$

$$\mathbf{m}_i = \mathbf{p}_i + \mathbf{c}\left(\left(\mathbf{m} - \mathbf{p}_i\right) \cdot \mathbf{c}\right) \tag{3.11}$$

We pick two tangents of $\mathbf{c}$, so that $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ forms a basis, and use them to define the position constraint:

$$\mathrm{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = \begin{bmatrix} (\mathbf{m}_b - \mathbf{m}_a) \cdot \mathbf{a} \\ (\mathbf{m}_b - \mathbf{m}_a) \cdot \mathbf{b} \end{bmatrix} \tag{3.12}$$

Taking the derivative gives us two velocity constraints of the form:

$$\dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = (\mathbf{v}_b + \boldsymbol{\omega}_b \times \mathbf{r}_b - \mathbf{v}_a - \boldsymbol{\omega}_a \times \mathbf{r}_a) \cdot \mathbf{n}$$

Where in this case $\mathbf{r}_i = \mathbf{m}_i - \mathbf{x}_i$ and $\mathbf{n} = \{\mathbf{a}, \mathbf{b}\}$. This leaves us with the same equation as (3.3).

3.2 Angular Constraints

This section describes a number of angular constraints. With angular constraints we mean constraints that primarily constrain rotational aspects between bodies.

3.2.1 Hinge constraint

The hinge constraint allows for 1 angular degree of freedom between bodies a and b, keeping the hinge axes of the bodies parallel.

We define the hinge axis $\mathbf{c}_i$ for body i in the local space of body i. We pick two tangents so that $\{\mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i\}$ forms a basis. The constraint ensures that the relative rotation about all axes other than $\mathbf{c}$ is zero:

$$\begin{aligned} \mathbf{R}_a\mathbf{c}_a \cdot \mathbf{R}_b\mathbf{a}_b &= 0 \\ \mathbf{R}_a\mathbf{c}_a \cdot \mathbf{R}_b\mathbf{b}_b &= 0 \end{aligned} \tag{3.13}$$

Where $\mathbf{R}_i$ is the local to world rotation matrix for body i.

We define the following for convince:

$$\begin{aligned} \mathbf{d} &= \mathbf{R}_b\mathbf{a}_b \\ \mathbf{e} &= \mathbf{R}_b\mathbf{b}_b \\ \mathbf{f} &= \mathbf{R}_a\mathbf{c}_a \end{aligned} \tag{3.14}$$

Now we can write the position constraint as:[8]

$$\mathrm{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = \begin{bmatrix} \mathbf{f} \cdot \mathbf{d} \\ \mathbf{f} \cdot \mathbf{e} \end{bmatrix} \tag{3.15}$$

Note that this will not work as expected with errors of 90° or more as $\mathbf{f} \cdot \mathbf{d} = \mathbf{f} \cdot -\mathbf{d}$ when $\mathbf{f}$ and $\mathbf{d}$ are perpendicular (See Sections 3.2.2 and 3.2.4 for hinge constraints without this problem).

The velocity constraint of the first row is obtained as follows:

$$\begin{aligned} \dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= \frac{d}{dt}(\mathbf{f} \cdot \mathbf{d}) \\ &= \dot{\mathbf{f}} \cdot \mathbf{d} + \mathbf{f} \cdot \dot{\mathbf{d}} \\ &= (\boldsymbol{\omega}_a \times \mathbf{f}) \cdot \mathbf{d} + \mathbf{f} \cdot (\boldsymbol{\omega}_b \times \mathbf{d}) \\ &= \boldsymbol{\omega}_a \cdot (\mathbf{f} \times \mathbf{d}) + \boldsymbol{\omega}_b \cdot (\mathbf{d} \times \mathbf{f}) \\ \mathbf{J}\mathbf{v} &= \begin{bmatrix} 0 \\ (\mathbf{f} \times \mathbf{d}) \\ 0 \\ (\mathbf{d} \times \mathbf{f}) \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{v}_a \\ \boldsymbol{\omega}_a \\ \mathbf{v}_b \\ \boldsymbol{\omega}_b \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ -(\mathbf{d} \times \mathbf{f}) \\ 0 \\ (\mathbf{d} \times \mathbf{f}) \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{v}_a \\ \boldsymbol{\omega}_a \\ \mathbf{v}_b \\ \boldsymbol{\omega}_b \end{bmatrix} \end{aligned} \tag{3.16}$$

Note that since $\mathbf{d}$, $\mathbf{e}$, and $\mathbf{f}$ are directions, no position or linear velocity is present in any of these equations.

We can obtain the effective mass by filling in Equation (2.6):

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\mathsf{T}} = \left(\mathbf{I}_a \left(-\mathbf{d} \times \mathbf{f}\right)^{-1}\right) \cdot \left(-\mathbf{d} \times \mathbf{f}\right) + \left(\mathbf{I}_b \left(\mathbf{d} \times \mathbf{f}\right)^{-1}\right) \cdot \left(\mathbf{d} \times \mathbf{f}\right) \tag{3.17}$$

The second row of the $\mathbf{J}$ can be derived similarly.

3.2.2 Double position hinge constraint

A hinge constraint can be constructed with two position constraints (from section 3.1.3) if the anchor points are picked along the hinge axis. This will effectively constrain the rotation to the hinge axis, and can recover from errors $\geq 90°$. However convergence is slow when errors are introduced since the two position constraints will counter each others angular correction. Every time the constraint corrects one point by rotating and translating the bodies, the other constraint will be broken again.

Convergence for translation errors improves when the two constrained points are picked closer together, however the precision for the rotation degrades. Convergence for rotational errors improves when the two points are picked far apart, and the linear precision degrades similarly.

Hanging chains tend to curve when the solver always iterates on the points in the same order. This is because the last of the two position constraints will always be better satisfied than the first.

Convergence is improved when both points are not picked symmetric, and one of the points is the point on the hinge axis closest to the center of mass of one of the bodies. This becomes quite similar to the constraint from Section 3.2.1, as now one constraint is primarily focusing on the satisfying the linear constraint (the closest point) and the other is focusing on rotation.

3.2.3 Quaternion constraint

This constraint keeps the relative orientation between two bodies constant, removing 3 degrees of freedom.

We begin by calculating the relative orientation $\mathbf{q}_r$ between bodies a and b in world space:

$$\begin{aligned} \mathbf{q}_{rw}\mathbf{q}_a &= \mathbf{q}_b \\ \mathbf{q}_{rw} &= \mathbf{q}_b\overline{\mathbf{q}}_a \end{aligned} \tag{3.18}$$

Where $\mathbf{q}_{rw}$ describes a rotation from $\mathbf{q}_a$ to $\mathbf{q}_b$.

We want to compare this rotation with an initial state $\overline{\mathbf{q}}_0$. The constraint forces both bodies to keep the same *relative* rotation, meaning that the rotation axis (imaginary part) of $\mathbf{q}_{rw}$ can change. This is not the case if both the initial state and the current relative rotation $\mathbf{q}_r$ are calculated in body space:

$$\begin{aligned} \mathbf{q}_r &= \overline{\mathbf{q}}_a\mathbf{q}_{rw}\mathbf{q}_a \\ &= \overline{\mathbf{q}}_a\mathbf{q}_b\overline{\mathbf{q}}_a\mathbf{q}_a \\ &= \overline{\mathbf{q}}_a\mathbf{q}_b \end{aligned} \tag{3.19}$$

Note that the result is the same in both the space of body a and b.

We calculate the initial relative orientation $\mathbf{q}_0$ when we create the constraint similarly:

$$\begin{aligned} \mathbf{q}_0 &= \overline{\mathbf{q}}_{a0} \left(\mathbf{q}_{b0} \overline{\mathbf{q}}_{a0} \right) \mathbf{q}_{a0} \\ \mathbf{q}_0 &= \overline{\mathbf{q}}_{a0} \mathbf{q}_{b0} \end{aligned} \tag{3.20}$$

We write the position constraint as follows:

$$\begin{aligned} \mathrm{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = \mathbf{q}_r &= \mathbf{q}_0 \\ &= \mathbf{q}_r \overline{\mathbf{q}}_0 = [1, 0, 0, 0] \end{aligned} \tag{3.21}$$

We want a position constraint with a range of $\mathbb{R}^3$. We can achieve this by only using the imaginary part of the quaternion:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \mathrm{c}(\ldots) &= \mathbf{A} \left(\mathbf{q}_r \overline{\mathbf{q}}_0 \right) = \mathbf{0} \end{aligned} \tag{3.22}$$

Where the multiplication with $\mathbf{A}$ treats the quaternion as an $\mathbb{R}^4$ column vector.

Now we take the derivative to obtain the velocity constraint:

$$\begin{aligned} \dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= \frac{d}{dt} \left(\mathbf{A} \left(\mathbf{q}_r \overline{\mathbf{q}}_0 \right) \right) \\ &= \frac{d}{dt} \left(\mathbf{A} \left(\overline{\mathbf{q}}_a \mathbf{q}_b \overline{\mathbf{q}}_0 \right) \right) \\ &= \mathbf{A} \left(\frac{d}{dt} \overline{\mathbf{q}}_a \mathbf{q}_b \overline{\mathbf{q}}_0 + \overline{\mathbf{q}}_a \frac{d}{dt} \mathbf{q}_b \overline{\mathbf{q}}_0 \right) \\ &= \mathbf{A} \left(\frac{1}{2} \left(\overline{\boldsymbol{\omega}_a \mathbf{q}_a} \right) \mathbf{q}_b \overline{\mathbf{q}}_0 + \overline{\mathbf{q}}_a \frac{1}{2} \left(\boldsymbol{\omega}_b \mathbf{q}_b \right) \overline{\mathbf{q}}_0 \right) \\ &= \mathbf{A} \frac{1}{2} \left(\overline{\mathbf{q}}_a \overline{\boldsymbol{\omega}}_a \mathbf{q}_b \overline{\mathbf{q}}_0 + \overline{\mathbf{q}}_a \boldsymbol{\omega}_b \mathbf{q}_b \overline{\mathbf{q}}_0 \right) \end{aligned}$$

Note that $\boldsymbol{\omega}$ is used as a pure imaginary quaternion, so $\overline{\boldsymbol{\omega}} = -\boldsymbol{\omega}$.

$$\begin{aligned} \dot{\mathrm{c}}(\ldots) &= \mathbf{A} \frac{1}{2} \left(-\overline{\mathbf{q}}_a \boldsymbol{\omega}_a \mathbf{q}_b \overline{\mathbf{q}}_0 + \overline{\mathbf{q}}_a \boldsymbol{\omega}_b \mathbf{q}_b \overline{\mathbf{q}}_0 \right) \\ &= \mathbf{A} \frac{1}{2} \left(\overline{\mathbf{q}}_a \boldsymbol{\omega}_b \mathbf{q}_b \overline{\mathbf{q}}_0 - \overline{\mathbf{q}}_a \boldsymbol{\omega}_a \mathbf{q}_b \overline{\mathbf{q}}_0 \right) \end{aligned}$$

We now rewrite this using matrices constructed from quaternions as described in Section A.3.4.

$$\begin{aligned} \dot{\mathrm{c}}(\ldots) &= \mathbf{A} \frac{1}{2} \left(\mathrm{M_L} \left(\overline{\mathbf{q}}_a \right) \mathrm{M_R} \left(\mathbf{q}_b \overline{\mathbf{q}}_0 \right) \boldsymbol{\omega}_b - \mathrm{M_L} \left(\overline{\mathbf{q}}_a \right) \mathrm{M_R} \left(\mathbf{q}_b \overline{\mathbf{q}}_0 \right) \boldsymbol{\omega}_a \right) \\ &= \mathbf{A} \, \mathrm{M_L} \left(\overline{\mathbf{q}}_a \right) \mathrm{M_R} \left(\mathbf{q}_b \overline{\mathbf{q}}_0 \right) \frac{1}{2} \left(\boldsymbol{\omega}_b - \boldsymbol{\omega}_a \right) \end{aligned}$$

We write this with $\boldsymbol{\omega}$ as $\mathbb{R}^3$ vectors again using the $\mathbf{A}$ matrix.

$$\mathbf{A}^\top \boldsymbol{\omega}_v = \boldsymbol{\omega}_q$$
$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_x \\ \boldsymbol{\omega}_y \\ \boldsymbol{\omega}_z \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{\omega}_x \\ \boldsymbol{\omega}_y \\ \boldsymbol{\omega}_z \end{bmatrix} \tag{3.23}$$

Giving us:

$$\dot{\mathrm{c}}(\ldots) = \mathbf{A}\,\mathrm{M_L}\left(\overline{\mathbf{q}}_a\right)\mathrm{M_R}\left(\mathbf{q}_b\overline{\mathbf{q}}_0\right)\mathbf{A}^\top \frac{1}{2}\left(\boldsymbol{\omega}_b - \boldsymbol{\omega}_a\right)$$

Splitting off the velocities gives us a 3×12 Jacobian:

$$\mathbf{J}\mathbf{v} = \begin{bmatrix} \left(-\frac{1}{2}\mathbf{A}\,\mathrm{M_L}\left(\overline{\mathbf{q}}_a\right)\mathrm{M_R}\left(\mathbf{q}_b\overline{\mathbf{q}}_0\right)\mathbf{A}^\top\right)^\top \\ \left(\ \ \frac{1}{2}\mathbf{A}\,\mathrm{M_L}\left(\overline{\mathbf{q}}_a\right)\mathrm{M_R}\left(\mathbf{q}_b\overline{\mathbf{q}}_0\right)\mathbf{A}^\top\right)^\top \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{\omega}_a \\ \boldsymbol{\omega}_b \end{bmatrix} \tag{3.24}$$

Note that $\det(\mathbf{J})$ will approach 0 as the angle between $\mathbf{q}_a$ and $\mathbf{q}_b\overline{\mathbf{q}}_0$ approaches 180°. This happens because there are infinite axes for the shortest rotation between two directions with an angle of 180°. This case should be handled by picking an arbitrary axis to rotate around until $\det(\mathbf{J})$ becomes big enough again to calculate the effective mass.

3.2.4 Quaternion hinge constraint

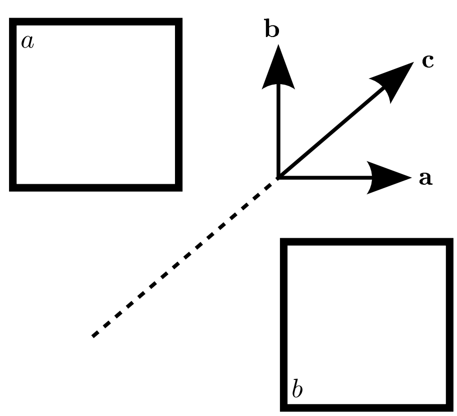


Figure 3.5: Bodies a and b are connected with a hinge constraint limiting the relative rotation to be along the hinge axis $\mathbf{c}$.

Here we describe a hinge constraint based on the quaternion constraint from Section 3.2.3. Bodies a and b are limited to only rotate around hinge axis $\mathbf{c}$ (see figure 3.5). We define $\mathbf{c}$ in the local space of body a, and we pick two tangents so that $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ forms a basis.

We calculate the initial rotation difference $\mathbf{q}_0$, to let both bodies rotate around a different local axis. $\mathbf{q}_0$ is the shortest rotation from the local hinge axes of body a to the local axis of body b, which can be calculated as described in Section A.3.6.

We modify the position constraint from (3.22) to only constrain rotation around $\mathbf{a}$ and $\mathbf{b}$:

$$\mathrm{c}(\ldots) = \mathbf{A}\left(\mathbf{q}_r\overline{\mathbf{q}}_0\right)\begin{bmatrix}\mathbf{a}^\top \\ \mathbf{b}^\top\end{bmatrix} = \mathbf{0} \tag{3.25}$$

We solve this as two separate constraints starting with $\mathbf{a}$. Taking the derivative gives us the velocity constraint:

$$\dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = \frac{d}{dt}\left(\mathbf{A}\left(\mathbf{q}_r\overline{\mathbf{q}}_0\right)\right)\mathbf{a} \tag{3.26}$$

Note that $\mathbf{a}$ is in local space of body a and is treated as a constant.

The rest of the derivative follows the same steps as shown in Section A.3.6, giving us:

$$\dot{\mathrm{c}}(\ldots) = \mathbf{A}\,\mathrm{M_L}\left(\overline{\mathbf{q}}_a\right)\mathrm{M_R}\left(\mathbf{q}_b\overline{\mathbf{q}}_0\right)\mathbf{A}^\top\frac{1}{2}\left(\boldsymbol{\omega}_b - \boldsymbol{\omega}_a\right)\mathbf{a} \tag{3.27}$$

$$\mathbf{jv} = \begin{bmatrix}\left(-\frac{1}{2}\mathbf{A}\,\mathrm{M_L}\left(\overline{\mathbf{q}}_a\right)\mathrm{M_R}\left(\mathbf{q}_b\overline{\mathbf{q}}_0\right)\mathbf{A}^\top\mathbf{a}\right)^\top \\ \left(\ \frac{1}{2}\mathbf{A}\,\mathrm{M_L}\left(\overline{\mathbf{q}}_a\right)\mathrm{M_R}\left(\mathbf{q}_b\overline{\mathbf{q}}_0\right)\mathbf{A}^\top\mathbf{a}\right)^\top\end{bmatrix}^\top\begin{bmatrix}\boldsymbol{\omega}_a \\ \boldsymbol{\omega}_b\end{bmatrix} \tag{3.28}$$

The constraint for axis $\mathbf{b}$ can be derived similarly.

3.2.5 Quaternion hinge limit

Here we describe how the rotation angle of the hinge from Section 3.2.4 can be limited to an interval.

We use an equation similar to (3.22) to calculate our rotation along the hinge axis:

$$\mathrm{c_{lim}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = \mathbf{A}\left(\mathbf{q}_r\overline{\mathbf{q}}_0\right)\cdot\mathbf{c} \tag{3.29}$$

Note that this is also the position constraint for the still unconstrained hinge axis $\mathbf{c}$.

We do not want to lock this axis, but only limit rotation about this axis to an interval:

$$l_{\mathrm{min}} < \mathrm{c_{lim}}(\ldots) < l_{\mathrm{max}} \tag{3.30}$$

Where l_{min} and l_{max} is the interval which can be calculated as follows:

$$\begin{aligned} l_{\mathrm{min}} &= \sin\left(\frac{1}{2}l_{\text{angle min}}\right) \\ l_{\mathrm{max}} &= \sin\left(\frac{1}{2}l_{\text{angle max}}\right) \end{aligned} \tag{3.31}$$

Where $l_{\text{angle min}}$ and $l_{\text{angle max}}$ are the angles between the limits and $\mathbf{b}$. The $\sin\left(\frac{1}{2}\text{ang}\right)$ is derived using (A.10):

$$\begin{aligned} c_{\text{lim}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= \mathbf{A}\left(\mathbf{q}_r\overline{\mathbf{q}}_0\right) \cdot \mathbf{c} \\ &= \left(\mathbf{q}_r\overline{\mathbf{q}}_0\right)_i \cdot \mathbf{c} \\ &= \sin\left(\frac{1}{2}\theta\right)\mathbf{d} \cdot \mathbf{c} \end{aligned}$$

Where $\mathbf{d}$ is the rotation axis and θ the angle of $\mathbf{q}_r\overline{\mathbf{q}}_0$. If we assume that the constraints locking the $\mathbf{a}$ and $\mathbf{b}$ axes are satisfied then we know that $\mathbf{d}$ and $\mathbf{c}$ are parallel, giving us:

$$c_{\text{lim}}(\ldots) = \sin\left(\frac{1}{2}\theta\right) \tag{3.32}$$

With (3.29) we calculate the rotation component around the $\mathbf{c}$ axis in half sine space, which we can use to compare with $l_{\min}$ and $l_{\max}$.

Each frame we check if $c_{\text{lim}}(\ldots)$ lies in our interval. If it does then we do not have to enforce the limit constraints. We solve the violated limit if $c_{\text{lim}}(\ldots)$ is outside our interval. The Jacobian of this constraint is similarly derived as (3.28). The position error is calculated as follows:

$$\begin{aligned} b_{\min} &= c_{\text{lim}}(\ldots) - l_{\min} \\ b_{\max} &= c_{\text{lim}}(\ldots) - l_{\max} \end{aligned} \tag{3.33}$$

Note that the sign of $\mathbf{q}_r\overline{\mathbf{q}}_0$ is important since we are comparing only the imaginary part. To avoid problems we negate the quaternion if its real component is negative. This ensures that our Jacobian always points in the same direction.

We clamp the reaction impulses of the limit constraints so that they can only push the object towards our interval:

$$\begin{aligned} 0 < \lambda_{\min} < +\infty \\ -\infty < \lambda_{\max} < 0 \end{aligned} \tag{3.34}$$

We can not pick an arbitrary basis if limits are used as the limit angle is relative to $\mathbf{b}$. The limits might get corrected in the wrong direction if the basis is picked poorly. If the limit is between $\mathbf{v}_1$ and $\mathbf{v}_2$ from figure 3.6, for example, and the limit is violated anticlockwise so that the hinge is rotated past 180° point, it will be corrected the wrong way. It is best to pick a basis so that $\mathbf{b}$ bisects the limit like $\mathbf{v}_3$ and $\mathbf{v}_4$ from figure 3.6.

We can exploit the fact that quaternions can store 720° rotations to let our limits go past 180° and avoid wrong way corrections when limits are broken past 180°. We need to modify the output of the position constraint to be able to do this:

$$c_{\text{lim2}}(\ldots) = \begin{cases} c_{\text{lim}}(\ldots) & \text{for } \left(\mathbf{q}_r\overline{\mathbf{q}}_0\right)_w > 0 \\ 2 - c_{\text{lim}}(\ldots) & \text{for } \left(\mathbf{q}_r\overline{\mathbf{q}}_0\right)_w < 0 \wedge c_{\text{lim}}(\ldots) < 0 \\ -2 - c_{\text{lim}}(\ldots) & \text{for } \left(\mathbf{q}_r\overline{\mathbf{q}}_0\right)_w < 0 \wedge c_{\text{lim}}(\ldots) > 0 \end{cases} \tag{3.35}$$

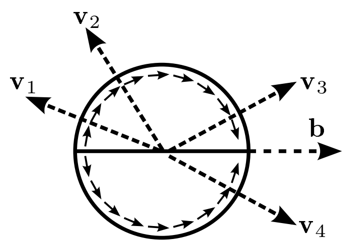


Figure 3.6: This diagram shows in which direction the limit will be corrected.

Figure 3.7 shows the plot of $c_{\text{lim}}(\ldots)$ in black and $c_{\text{lim2}}(\ldots)$ in green, red, and blue for the different cases respectively. The figure also shows the plot of $(\mathbf{q}_r\overline{\mathbf{q}}_0)_w$ in gray. Note that the domain for which (3.35) works as expected is $(-3\pi, 3\pi)$.

Note that $\mathbf{j}$ should still be flipped when $(\mathbf{q}_r\overline{\mathbf{q}}_0)_w < 0$ to keep correcting the limits in the correct direction. Also note that it is now possible to have the system in a seemingly correct configuration with violated limits because we use the full 720° of the quaternion to specify the limit. This can be resolved by flipping one of the body's quaternions at initialization, ensuring the system always starts in a correct configuration.

3.2.6 Cone limit

This constraint limits the relative rotation, excluding twist about $\mathbf{c}$, between bodies a and b to ϕ (See Figure 3.8).

We can calculate this angle using the dot product:

$$\begin{aligned} \theta &= \text{acos}\left(\mathbf{c}_a \cdot \mathbf{c}_b\right) \\ \cos\left(\theta\right) &= \mathbf{c}_a \cdot \mathbf{c}_b \end{aligned} \tag{3.36}$$

Where $\mathbf{c}_i$ is a reference vector in world space for body i.

We start with the following position constraint:

$$\begin{aligned} \text{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= \mathbf{c}_a \cdot \mathbf{c}_b = \cos\left(\phi\right) \\ &= \mathbf{c}_a \cdot \mathbf{c}_b - \cos\left(\phi\right) = 0 \end{aligned} \tag{3.37}$$

Taking the derivative gives us the velocity constraint:

$$\begin{aligned} \dot{\text{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= \dot{\mathbf{c}_a}\mathbf{c}_b + \mathbf{c}_a\dot{\mathbf{c}}_b = 0 \\ &= (\boldsymbol{\omega}_a \times \mathbf{c}_a) \cdot \mathbf{c}_b + \mathbf{c}_a \cdot (\boldsymbol{\omega}_b \times \mathbf{c}_b) \\ &= (\mathbf{c}_b \times \mathbf{c}_a) \cdot \boldsymbol{\omega}_b - (\mathbf{c}_b \times \mathbf{c}_a) \cdot \boldsymbol{\omega}_a \end{aligned} \tag{3.38}$$

$$\mathbf{jv} = \begin{bmatrix} -\mathbf{c}_b \times \mathbf{c}_a \\ \mathbf{c}_b \times \mathbf{c}_a \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \boldsymbol{\omega}_a \\ \boldsymbol{\omega}_b \end{bmatrix} \tag{3.39}$$

Since this is a limit, the constraint can be set up as inequality constraint with $0 < \lambda < +\infty$ and it only needs to be enforced when $\mathbf{c}_a \cdot \mathbf{c}_b < \cos(\phi)$

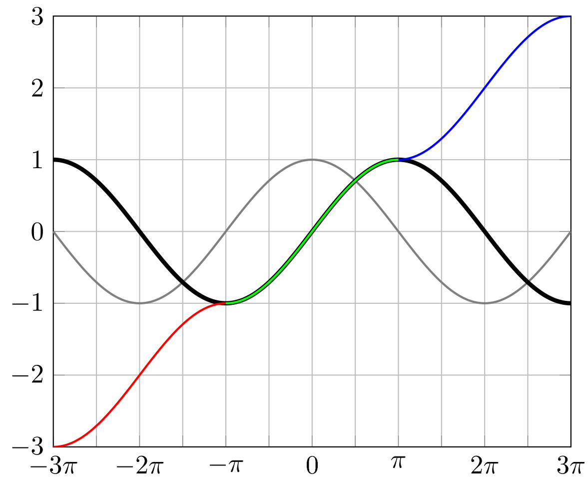


Figure 3.7: Plot of the quaternion position constraint (black), the extended limit position constraint (green/red/blue), and the real part of the quaternion (gray).

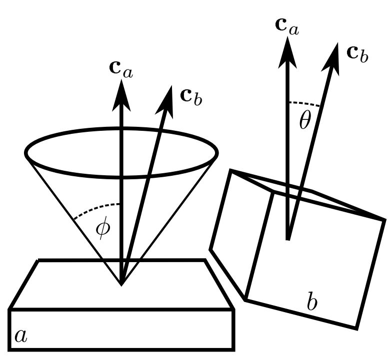


Figure 3.8: The relative rotation between $\mathbf{c}_a$ and $\mathbf{c}_b$ is limited to angle ϕ.

3.2.7 Wedge limit

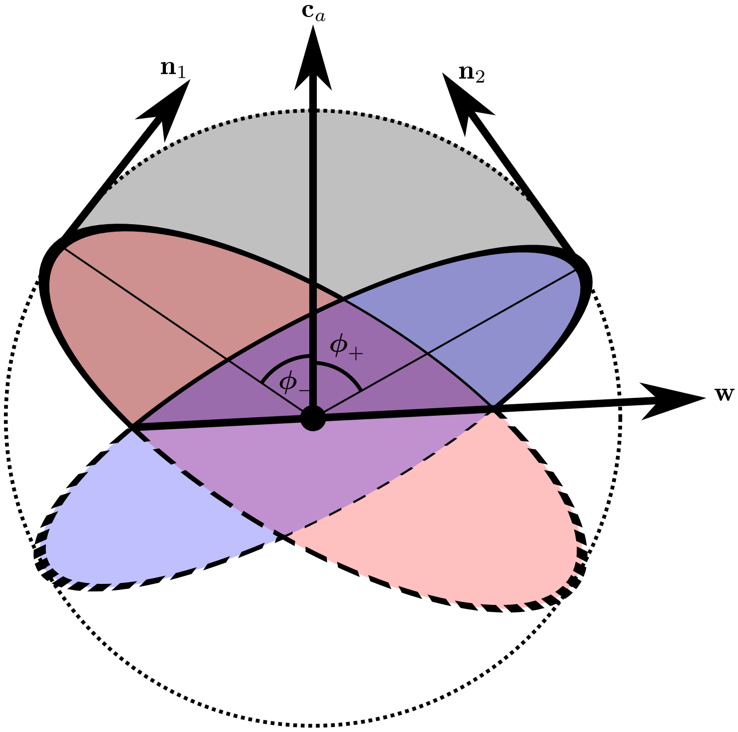


Figure 3.9: Wedge limit.

This constraint limits the relative orientation between two bodies, using $\mathbf{c}_a$ and $\mathbf{c}_b$ as reference directions, to a wedge of a sphere. We pick two planes, with normals $\mathbf{n}_1$ and $\mathbf{n}_2$, on which the flat sides of the wedge lie. Axis $\mathbf{c}_b$ is constrained to the lune of the wedge defined by the normals (see Figure 3.9). This can be expressed with an inequality constraint, forcing $\mathbf{c}_b$ to be on or above the planes:

$$\mathrm{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = \begin{bmatrix} \mathbf{c}_b \cdot \mathbf{n}_1 \\ \mathbf{c}_b \cdot \mathbf{n}_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.40}$$

Taking the derivative gives us the velocity constraints.

$$\begin{aligned} \dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= \frac{d}{dt}\mathbf{c}_b \cdot \mathbf{n} + \mathbf{c}_b \cdot \frac{d}{dt}\mathbf{n} \\ &= (\boldsymbol{\omega}_b \times \mathbf{c}_b) \cdot \mathbf{n} + \mathbf{c}_b \cdot (\boldsymbol{\omega}_a \times \mathbf{n}) \\ &= (\mathbf{c}_b \times \mathbf{n}) \cdot \boldsymbol{\omega}_b - (\mathbf{c}_b \times \mathbf{n}) \cdot \boldsymbol{\omega}_a \\ \mathbf{jv} &= \begin{bmatrix} -\mathbf{c} \times \mathbf{n} \\ \mathbf{c} \times \mathbf{n} \end{bmatrix}^\intercal \begin{bmatrix} \boldsymbol{\omega}_a \\ \boldsymbol{\omega}_b \end{bmatrix} \end{aligned} \tag{3.41}$$

We can construct the planes from the limit angles ϕ_- and ϕ_+ given the limits wedge axis $\mathbf{w}$. We first rotate $\mathbf{c}_a$ over axis $\mathbf{w}$ by ϕ to obtain $\mathbf{v}$ lying on the plane, then we calculate the normal from $\mathbf{v}$ and $\mathbf{w}$.

$$\mathbf{q_w} = \mathrm{q}\odot(\mathbf{w}, \phi) \tag{3.42}$$

$$\mathbf{v} = \mathbf{q_w} \mathbf{c}_a \overline{\mathbf{q}}_\mathbf{w} \tag{3.43}$$

$$\mathbf{n} = \frac{\mathbf{v} \times \mathbf{w}}{|\mathbf{v} \times \mathbf{w}|} \tag{3.44}$$

Where $\mathrm{q}\odot(\ldots)$ is defined in Section A.3.1. The calculation of the normal is similar for both the minimum and maximum limit, aside from a sign change in the cross product.

The constraint as is can not have limit angles $\phi_+ - \phi_- > 180°$ as the wrong side of the planes will start to limit. We can check the sign of $(\mathbf{c}_a \times \mathbf{w}) \cdot \mathbf{c}_b$ to determine which of the limits should be considered to remove this limitation. However this introduces instability when $\mathbf{c}_b$ is resting in the corner of the wedge (along $\mathbf{w}$), as only one of the limits can be active. Convergence for acute wedges can be improved by adding an extra plane limit constraint with normal $\mathbf{c}_a$ when both the minimum and maximum limits are violated.

3.2.8 Angular limit

Here we describe a common constraint used to limit angular motion. This limit should only be used with < 3 angular degrees of freedom as noted below.

We define two local bases for the bodies:

$$\begin{aligned} \text{basis for body } a &= \{\mathbf{a}_a, \mathbf{b}_a, \mathbf{c}_a\} \\ \text{basis for body } b &= \{\mathbf{a}_b, \mathbf{b}_b, \mathbf{c}_b\} \end{aligned} \tag{3.45}$$

We limit the relative rotation θ between bodies a and b about $\mathbf{c}_a$ so that $\phi_- < \theta < \phi_+$. This gives the following position constraint for the max limit:

$$\begin{aligned} \mathrm{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= \theta < \phi_+ \\ &= \theta - \phi_+ < 0 \end{aligned} \tag{3.46}$$

Where $\theta = \mathrm{atan2}(\mathbf{b}_a \cdot \mathbf{a}_b, \mathbf{a}_a \cdot \mathbf{a}_b)$ as shown in Section A.2.1.

For the derivative we use:

$$\begin{aligned} \dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= \frac{d}{dt}\theta \\ &= \mathbf{c}_a \cdot (\boldsymbol{\omega}_b - \boldsymbol{\omega}_a) \\ \mathbf{jv} &= \begin{bmatrix} -\mathbf{c}_a \\ \mathbf{c}_a \end{bmatrix}^\mathsf{T} \begin{bmatrix} \boldsymbol{\omega}_a \\ \boldsymbol{\omega}_b \end{bmatrix} \end{aligned} \tag{3.47}$$

The min limit is obtained similarly.

Note that this limit constraint has some problems.

The calculation of θ is imprecise when $\mathbf{a}_b$ becomes parallel to $\mathbf{c}_a$. This makes the constraint unsuitable for many situations where the limit angle about the other axes approaches or passes 90°.

Compared to the wedge limit, this constraint does not correct errors along the shortest path when $\mathbf{a}_b$ does not lie on the $\mathbf{a}_a\mathbf{b}_a$ plane. This introduce unnecessary rotation about $\mathbf{c}_a$ in systems with 3 angular degrees of freedom, and can cause poor convergence in systems with 2 angular degrees of freedom.

3.2.9 Universal joint

In this section we describe a universal joint constraint. The universal joint uses a pair of hinges with perpendicular hinge axis to transfer angular motion between axes under an angle. The hinge along $\mathbf{z}$ is fixed to the frame of reference of body a, and the hinge along $\mathbf{y}$ is fixed to the frame of body b (See Figure 3.10):

$$
\begin{aligned}
\mathbf{z} &= \mathbf{R}_a \mathbf{z}_a \\
\mathbf{y} &= \mathbf{R}_b \mathbf{y}_b
\end{aligned}
$$

The two hinges are connected by a spider which keeps $\mathbf{z}$ and $\mathbf{y}$ right-angled. This relation can be described in a position constraint as follows:

$$
\mathrm{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = \mathbf{z} \cdot \mathbf{y} = 0
$$

Taking the derivative gives us the velocity constraint:

$$
\begin{aligned}
\dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= \frac{d}{dt}\mathbf{z} \cdot \mathbf{y} + \mathbf{z} \cdot \frac{d}{dt}\mathbf{y} \\
&= (\boldsymbol{\omega}_a \times \mathbf{z}) \cdot \mathbf{y} + \mathbf{z} \cdot (\boldsymbol{\omega}_b \times \mathbf{y}) \\
&= (\mathbf{z} \times \mathbf{y}) \cdot \boldsymbol{\omega}_a - (\mathbf{z} \times \mathbf{y}) \cdot \boldsymbol{\omega}_b \\
\mathbf{jv} &= \begin{bmatrix} (\mathbf{z} \times \mathbf{y}) \\ -(\mathbf{z} \times \mathbf{y}) \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \boldsymbol{\omega}_a \\ \boldsymbol{\omega}_b \end{bmatrix}
\end{aligned}
$$

3.2.10 Constant velocity joint

In this section we describe a constant velocity (CV) joint, also known as homokinetic joint, similar to a Rzeppa joint. Unlike the universal joint, this joint will keep the velocity on the connected rotation axes equal.

The position constraint is similar to the universal constraint, but uses an extra set of reference axes (see Figure 3.11):[10]

$$
\mathrm{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = \mathbf{y}_a \cdot \mathbf{z}_b - \mathbf{z}_a \cdot \mathbf{y}_b = 0 \tag{3.48}
$$

It can visually be inspected that this equation holds as long as both bodies have the same rotation along their own $\mathbf{x}$ axis.

Taking the derivative gives us the velocity constraint:

$$
\begin{aligned}
\dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= \dot{\mathbf{y}}_a \cdot \mathbf{z}_b + \mathbf{y}_a \cdot \dot{\mathbf{z}}_b - \dot{\mathbf{z}}_a \cdot \mathbf{y}_b - \mathbf{z}_a \cdot \dot{\mathbf{y}}_b \\
&= (\boldsymbol{\omega}_a \times \mathbf{y}_a) \cdot \mathbf{z}_b + \mathbf{y}_a \cdot (\boldsymbol{\omega}_b \times \mathbf{z}_b) - (\boldsymbol{\omega}_a \times \mathbf{z}_a) \cdot \mathbf{y}_b - \mathbf{z}_a \cdot (\boldsymbol{\omega}_b \times \mathbf{y}_b) \\
&= \boldsymbol{\omega}_a \cdot (\mathbf{y}_a \times \mathbf{z}_b) + \boldsymbol{\omega}_b \cdot (\mathbf{z}_b \times \mathbf{y}_a) - \boldsymbol{\omega}_a \cdot (\mathbf{z}_a \times \mathbf{y}_b) - \boldsymbol{\omega}_b \cdot (\mathbf{y}_b \times \mathbf{z}_a) \\
\mathbf{jv} &= \begin{bmatrix} \mathbf{y}_a \times \mathbf{z}_b - \mathbf{z}_a \times \mathbf{y}_b \\ \mathbf{z}_b \times \mathbf{y}_a - \mathbf{y}_b \times \mathbf{z}_a \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \boldsymbol{\omega}_a \\ \boldsymbol{\omega}_b \end{bmatrix} \\
&= \begin{bmatrix} -(\mathbf{z}_b \times \mathbf{y}_a - \mathbf{y}_b \times \mathbf{z}_a) \\ (\mathbf{z}_b \times \mathbf{y}_a - \mathbf{y}_b \times \mathbf{z}_a) \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \boldsymbol{\omega}_a \\ \boldsymbol{\omega}_b \end{bmatrix}
\end{aligned} \tag{3.49}
$$

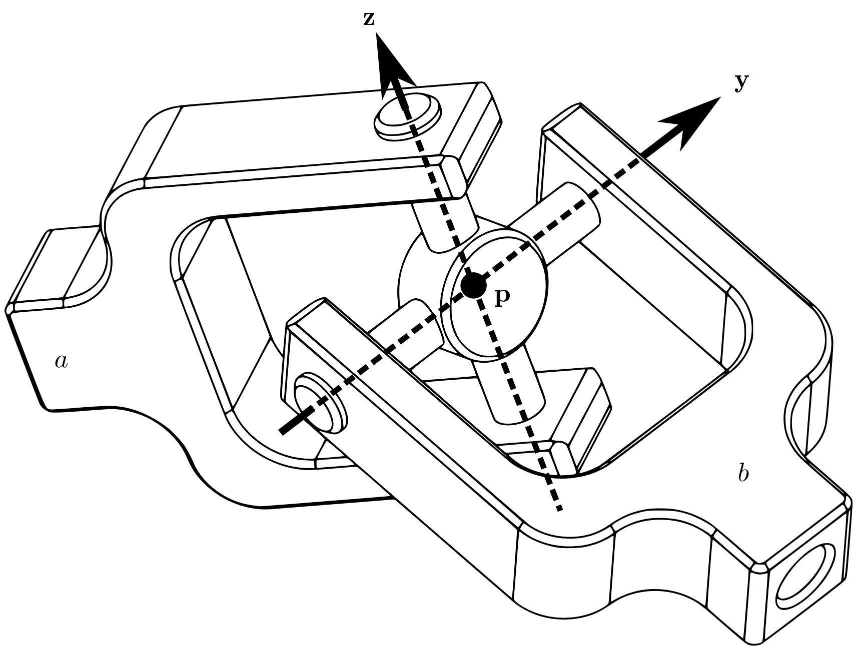


Figure 3.10: A universal constraint between bodies a and b. The orientation of b relative to a can be described by a sequence of two intrinsic rotations, first about $\mathbf{z}$ then about $\mathbf{y}$.

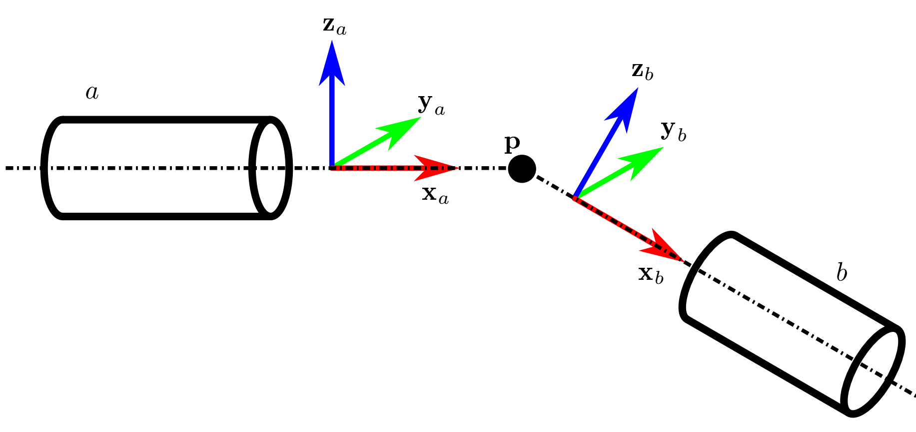


Figure 3.11: A constant velocity joint between bodies a and b. The bodies are connected at point $\mathbf{p}$. Body b is rotated 30° about $\mathbf{y}$ in this figure.

Note that $(\mathbf{z}_b \times \mathbf{y}_a - \mathbf{y}_b \times \mathbf{z}_a)$ gives us a vector that describes the normal of a plane that bisects $\mathbf{x}_a$ and $\mathbf{x}_b$ which is correct as the relative angular velocity $\boldsymbol{\omega}_b - \boldsymbol{\omega}_a$ must lie on this plane[2].

3.3 Unstable Angular Constraints

This section lists a number of angular constraints that have proven to be unstable. We include them in this document for reference and completeness.

3.3.1 Cross product single row hinge constraint

This constraint uses the properties of the cross product to constrain the relative orientation to be around a single axis. We know that $\mathbf{a} \times \mathbf{b} = \mathbf{0}$ as long as $\mathbf{a}$ and $\mathbf{b}$ are collinear. Note that this constraint is unstable as it is supposed to remove two degrees of freedom, however it has a Jacobian of only one row.

The position constraint is written as follows:

$$\mathrm{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = \frac{1}{2} \left|(\mathbf{R}_a \mathbf{c}_a) \times (\mathbf{R}_b \mathbf{c}_b)\right|^2 \tag{3.50}$$

Where $\mathbf{c}_i$ is the the hinge axis of body i in local space, and $\mathbf{R}_i$ is the local to world rotation matrix for body i. The $\frac{1}{2}$ and the square are added to simplify the derivative.

The velocity constraint becomes:

$$\dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = ((\mathbf{R}_a \mathbf{c}_a) \times (\mathbf{R}_b \mathbf{c}_b)) \cdot \frac{d}{dt} ((\mathbf{R}_a \mathbf{c}_a) \times (\mathbf{R}_b \mathbf{c}_b))$$

The derivative of the cross product:

$$\begin{aligned}
\frac{d}{dt} ((\mathbf{R}_a \mathbf{c}_a) \times (\mathbf{R}_b \mathbf{c}_b)) &= \frac{d}{dt} (\mathbf{R}_a \mathbf{c}_a) \times (\mathbf{R}_b \mathbf{c}_b) + (\mathbf{R}_a \mathbf{c}_a) \times \frac{d}{dt} (\mathbf{R}_b \mathbf{c}_b) \\
&= (\boldsymbol{\omega}_a \times \mathbf{r}_a) \times (\mathbf{R}_b \mathbf{c}_b) + (\mathbf{R}_a \mathbf{c}_a) \times (\boldsymbol{\omega}_b \times \mathbf{r}_b) \\
&= (\boldsymbol{\omega}_a \times \mathbf{r}_a) \times (\mathbf{R}_b \mathbf{c}_b) + (\mathbf{R}_a \mathbf{c}_a) \times (\mathbf{r}_b \times \boldsymbol{\omega}_b) \\
&= \boldsymbol{\omega}_a \times (\mathbf{r}_a \times (\mathbf{R}_b \mathbf{c}_b)) + ((\mathbf{R}_a \mathbf{c}_a) \times \mathbf{r}_b) \times \boldsymbol{\omega}_b \\
&= -(\mathbf{r}_a \times (\mathbf{R}_b \mathbf{c}_b)) \times \boldsymbol{\omega}_a + ((\mathbf{R}_a \mathbf{c}_a) \times \mathbf{r}_b) \times \boldsymbol{\omega}_b
\end{aligned}$$

Putting everything together:

$$\dot{\mathrm{c}}(\ldots) = ((\mathbf{R}_a \mathbf{c}_a) \times (\mathbf{R}_b \mathbf{c}_b)) \cdot (-(\mathbf{r}_a \times (\mathbf{R}_b \mathbf{c}_b)) \times \boldsymbol{\omega}_a + ((\mathbf{R}_a \mathbf{c}_a) \times \mathbf{r}_b) \times \boldsymbol{\omega}_b)$$

We continue with the following definitions to keep the equations short:

$$\begin{aligned}
\mathbf{d} &= (\mathbf{R}_a \mathbf{c}_a) \times (\mathbf{R}_b \mathbf{c}_b) \\
\mathbf{e} &= \mathbf{r}_a \times (\mathbf{R}_b \mathbf{c}_b) \\
\mathbf{f} &= (\mathbf{R}_a \mathbf{c}_a) \times \mathbf{r}_b
\end{aligned} \tag{3.51}$$

Making the velocity constraint:

$$\begin{aligned}
\dot{\mathrm{c}}(\ldots) &= \mathbf{d} \cdot (-\mathbf{e} \times \boldsymbol{\omega}_a + \mathbf{f} \times \boldsymbol{\omega}_b) \\
&= \mathbf{d} \cdot (-\mathbf{e} \times \boldsymbol{\omega}_a) + \mathbf{d} \cdot (\mathbf{f} \times \boldsymbol{\omega}_b) \\
&= (\mathbf{d} \times -\mathbf{e}) \cdot \boldsymbol{\omega}_a + (\mathbf{d} \times \mathbf{f}) \cdot \boldsymbol{\omega}_b \\
\mathbf{Jv} &= \begin{bmatrix} \mathbf{e} \times \mathbf{d} \\ \mathbf{d} \times \mathbf{f} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \boldsymbol{\omega}_a \\ \boldsymbol{\omega}_b \end{bmatrix}
\end{aligned} \tag{3.52}$$

We can obtain the effective mass by filling in Equation (2.6):

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^\mathsf{T} = \left(\mathbf{I}_a^{-1}(\mathbf{e} \times \mathbf{d})\right) \cdot (\mathbf{e} \times \mathbf{d}) + \left(\mathbf{I}_b^{-1}(\mathbf{d} \times \mathbf{f})\right) \cdot (\mathbf{d} \times \mathbf{f}) \tag{3.53}$$

3.3.2 Cross product matrix hinge constraint

Here we derive a similar constraint as in the previous section. We now use a 3 row Jacobian instead of a single row. We also show that this constraint is unsuitable for a block solver.

Instead of using the magnitude of the cross product, we use the whole cross product for the position constraint:

$$\mathrm{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) = (\mathbf{R}_a \mathbf{c}_a) \times (\mathbf{R}_b \mathbf{c}_b) \tag{3.54}$$

Where $\mathbf{c}_i$ is the the hinge axis of body i in local space, and $\mathbf{R}_i$ is the local to world rotation matrix for body i.

The velocity constraint becomes:

$$\begin{aligned}
\dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) =& \frac{d}{dt}(\mathbf{R}_a \mathbf{c}_a) \times (\mathbf{R}_b \mathbf{c}_b) + (\mathbf{R}_a \mathbf{c}_a) \times \frac{d}{dt}(\mathbf{R}_b \mathbf{c}_b) \\
=& (\boldsymbol{\omega}_a \times \mathbf{r}_a) \times (\mathbf{R}_b \mathbf{c}_b) + (\mathbf{R}_a \mathbf{c}_a) \times (\boldsymbol{\omega}_b \times \mathbf{r}_b) \\
=& (\boldsymbol{\omega}_a \times \mathbf{r}_a) \times (\mathbf{R}_b \mathbf{c}_b) + (\mathbf{R}_a \mathbf{c}_a) \times (\mathbf{r}_b \times \boldsymbol{\omega}_b) \\
=& \boldsymbol{\omega}_a \times (\mathbf{r}_a \times (\mathbf{R}_b \mathbf{c}_b)) + ((\mathbf{R}_a \mathbf{c}_a) \times \mathbf{r}_b) \times \boldsymbol{\omega}_b \\
=& -(\mathbf{r}_a \times (\mathbf{R}_b \mathbf{c}_b)) \times \boldsymbol{\omega}_a + ((\mathbf{R}_a \mathbf{c}_a) \times \mathbf{r}_b) \times \boldsymbol{\omega}_b \\
\mathbf{Jv} =& \begin{bmatrix} [(\mathbf{R}_b \mathbf{c}_b) \times \mathbf{r}_a]_\times^\mathsf{T} \\ [(\mathbf{R}_a \mathbf{c}_a) \times \mathbf{r}_b]_\times^\mathsf{T} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \boldsymbol{\omega}_a \\ \boldsymbol{\omega}_b \end{bmatrix}
\end{aligned} \tag{3.55}$$

We define for convenience:

$$\begin{aligned}
\mathbf{S}_a &= [(\mathbf{R}_b \mathbf{c}_b) \times \mathbf{r}_a]_\times \\
\mathbf{S}_b &= [(\mathbf{R}_a \mathbf{c}_a) \times \mathbf{r}_b]_\times
\end{aligned} \tag{3.56}$$

This gives us the following effective mass:

$$\begin{aligned}
\mathbf{J}\mathbf{M}^{-1} &= \begin{bmatrix} \mathbf{S}_a & \mathbf{S}_b \end{bmatrix} \begin{bmatrix} \mathbf{I}_a^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_b^{-1} \end{bmatrix} \\
\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^\mathsf{T} &= \begin{bmatrix} \mathbf{S}_a \mathbf{I}_a^{-1} & \mathbf{S}_b \mathbf{I}_b^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{S}_a^\mathsf{T} \\ \mathbf{S}_b^\mathsf{T} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{S}_a \mathbf{I}_a^{-1} \mathbf{S}_a^\mathsf{T} + \mathbf{S}_b \mathbf{I}_b^{-1} \mathbf{S}_b^\mathsf{T} \end{bmatrix}
\end{aligned} \tag{3.57}$$

Note that the effective mass matrix is singular as the cross product matrices are singular as well:

$$\begin{aligned}
\mathbf{d} &= (\mathbf{R}_b \mathbf{c}_b) \times \mathbf{r}_a \\
\mathbf{e} &= (\mathbf{R}_a \mathbf{c}_a) \times \mathbf{r}_b \\
\mathbf{S}_a &= \begin{bmatrix} 0 & -\mathbf{d}_z & \mathbf{d}_y \\ \mathbf{d}_z & 0 & -\mathbf{d}_x \\ -\mathbf{d}_y & \mathbf{d}_x & 0 \end{bmatrix} \\
\det(\mathbf{S}_a) &= (-\mathbf{d}_z \mathbf{d}_x \mathbf{d}_y + \mathbf{d}_y \mathbf{d}_z \mathbf{d}_x) = 0 \\
\det\left(\mathbf{S}_a \mathbf{I}_a \mathbf{S}_a^\mathsf{T}\right) &= \det(\mathbf{S}_a) \det(\mathbf{I}_a) \det\left(\mathbf{S}_a^\mathsf{T}\right) = 0
\end{aligned}$$

The determinant can be obtained similarly for the remaining part, giving:

$$\det\left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^\mathsf{T}\right) = 0 \tag{3.58}$$

This makes the constraint unsuitable for block solvers as the effective mass matrix is not invertible.

3.3.3 Single row quaternion constraint

This constraint forces the two bodies to keep the same orientation relative to each other, removing 3 degrees of freedom. This constraint is unstable as it uses a single row Jacobian, enabling it to correct the orientation along only one axis. This constraint is similar to the constraint described in Section 3.3.1 in this regard.

We write the position constraint as follows:

$$\begin{aligned}
\mathrm{c}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= \mathbf{q}_b \mathbf{q}_a^{-1} \cdot \mathbf{w} - 1 \\
&= \mathbf{q}_b \cdot \mathbf{q}_a - 1
\end{aligned} \tag{3.59}$$

Where $\mathbf{w} = [1, 0, 0, 0]$ and where dot $\cdot$ is the common vector dot product. Note that identity (A.14) was used.

Derivative of the two factors are:

$$\begin{aligned}
\frac{d}{dt}\mathbf{q}_a &= \left[0, \frac{1}{2}\boldsymbol{\omega}_a\right] \mathbf{q}_a \\
\frac{d}{dt}\mathbf{q}_b &= \left[0, \frac{1}{2}\boldsymbol{\omega}_b\right] \mathbf{q}_b
\end{aligned} \tag{3.60}$$

Making the velocity constraint the following:

$$\begin{aligned}
\dot{\mathrm{c}}(\mathbf{x}_a, \mathbf{q}_a, \mathbf{x}_b, \mathbf{q}_b) &= \frac{d}{dt}(\mathbf{q}_a \cdot \mathbf{q}_b) \\
&= \frac{d}{dt}\mathbf{q}_a \cdot \mathbf{q}_b + \mathbf{q}_a \cdot \frac{d}{dt}\mathbf{q}_b \\
&= \left[0, \frac{1}{2}\boldsymbol{\omega}_a\right] \mathbf{q}_a \cdot \mathbf{q}_b + \mathbf{q}_a \cdot \left[0, \frac{1}{2}\boldsymbol{\omega}_b\right] \mathbf{q}_b
\end{aligned}$$

Rewriting the first part:

$$\left[0, \frac{1}{2}\boldsymbol{\omega}_a\right] \mathbf{q}_a \cdot \mathbf{q}_b = \begin{bmatrix} (-\boldsymbol{\omega}_{ax}\mathbf{q}_{ax} - \boldsymbol{\omega}_{ay}\mathbf{q}_{ay} - \boldsymbol{\omega}_{az}\mathbf{q}_{az}) \\ (\boldsymbol{\omega}_{ax}\mathbf{q}_{aw} + \boldsymbol{\omega}_{ay}\mathbf{q}_{az} - \boldsymbol{\omega}_{az}\mathbf{q}_{ay}) \\ (-\boldsymbol{\omega}_{ax}\mathbf{q}_{az} + \boldsymbol{\omega}_{ay}\mathbf{q}_{aw} + \boldsymbol{\omega}_{az}\mathbf{q}_{ax}) \\ (\boldsymbol{\omega}_{ax}\mathbf{q}_{ay} - \boldsymbol{\omega}_{ay}\mathbf{q}_{ax} + \boldsymbol{\omega}_{az}\mathbf{q}_{aw}) \end{bmatrix} \cdot \mathbf{q}_b$$

$$= \begin{matrix} (-\boldsymbol{\omega}_{ax}\mathbf{q}_{ax} - \boldsymbol{\omega}_{ay}\mathbf{q}_{ay} - \boldsymbol{\omega}_{az}\mathbf{q}_{az})\,\mathbf{q}_{bw} + \\ (\boldsymbol{\omega}_{ax}\mathbf{q}_{aw} + \boldsymbol{\omega}_{ay}\mathbf{q}_{az} - \boldsymbol{\omega}_{az}\mathbf{q}_{ay})\,\mathbf{q}_{bx} + \\ (-\boldsymbol{\omega}_{ax}\mathbf{q}_{az} + \boldsymbol{\omega}_{ay}\mathbf{q}_{aw} + \boldsymbol{\omega}_{az}\mathbf{q}_{ax})\,\mathbf{q}_{by} + \\ (\boldsymbol{\omega}_{ax}\mathbf{q}_{ay} - \boldsymbol{\omega}_{ay}\mathbf{q}_{ax} + \boldsymbol{\omega}_{az}\mathbf{q}_{aw})\,\mathbf{q}_{bz} \end{matrix}$$

$$= \begin{matrix} -\boldsymbol{\omega}_{ax}\mathbf{q}_{ax}\mathbf{q}_{bw} - \boldsymbol{\omega}_{ay}\mathbf{q}_{ay}\mathbf{q}_{bw} - \boldsymbol{\omega}_{az}\mathbf{q}_{az}\mathbf{q}_{bw} + \\ \boldsymbol{\omega}_{ax}\mathbf{q}_{aw}\mathbf{q}_{bx} + \boldsymbol{\omega}_{ay}\mathbf{q}_{az}\mathbf{q}_{bx} - \boldsymbol{\omega}_{az}\mathbf{q}_{ay}\mathbf{q}_{bx} + \\ -\boldsymbol{\omega}_{ax}\mathbf{q}_{az}\mathbf{q}_{by} + \boldsymbol{\omega}_{ay}\mathbf{q}_{aw}\mathbf{q}_{by} + \boldsymbol{\omega}_{az}\mathbf{q}_{ax}\mathbf{q}_{by} + \\ \boldsymbol{\omega}_{ax}\mathbf{q}_{ay}\mathbf{q}_{bz} - \boldsymbol{\omega}_{ay}\mathbf{q}_{ax}\mathbf{q}_{bz} + \boldsymbol{\omega}_{az}\mathbf{q}_{aw}\mathbf{q}_{bz} \end{matrix}$$

$$= \begin{matrix} \boldsymbol{\omega}_{ax}\left(-\mathbf{q}_{ax}\mathbf{q}_{bw} + \mathbf{q}_{aw}\mathbf{q}_{bx} - \mathbf{q}_{az}\mathbf{q}_{by} + \mathbf{q}_{ay}\mathbf{q}_{bz}\right) + \\ \boldsymbol{\omega}_{ay}\left(-\mathbf{q}_{ay}\mathbf{q}_{bw} + \mathbf{q}_{az}\mathbf{q}_{bx} + \mathbf{q}_{aw}\mathbf{q}_{by} - \mathbf{q}_{ax}\mathbf{q}_{bz}\right) + \\ \boldsymbol{\omega}_{az}\left(-\mathbf{q}_{az}\mathbf{q}_{bw} - \mathbf{q}_{ay}\mathbf{q}_{bx} + \mathbf{q}_{ax}\mathbf{q}_{by} + \mathbf{q}_{aw}\mathbf{q}_{bz}\right) \end{matrix}$$

The second part can be rewritten similarly. Putting everything back together gives:

$$\dot{c}(\ldots) = \begin{matrix} \boldsymbol{\omega}_{ax}\left(-\mathbf{q}_{ax}\mathbf{q}_{bw} + \mathbf{q}_{aw}\mathbf{q}_{bx} - \mathbf{q}_{az}\mathbf{q}_{by} + \mathbf{q}_{ay}\mathbf{q}_{bz}\right) + \\ \boldsymbol{\omega}_{ay}\left(-\mathbf{q}_{ay}\mathbf{q}_{bw} + \mathbf{q}_{az}\mathbf{q}_{bx} + \mathbf{q}_{aw}\mathbf{q}_{by} - \mathbf{q}_{ax}\mathbf{q}_{bz}\right) + \\ \boldsymbol{\omega}_{az}\left(-\mathbf{q}_{az}\mathbf{q}_{bw} - \mathbf{q}_{ay}\mathbf{q}_{bx} + \mathbf{q}_{ax}\mathbf{q}_{by} + \mathbf{q}_{aw}\mathbf{q}_{bz}\right) \\ + \\ \boldsymbol{\omega}_{bx}\left(-\mathbf{q}_{bx}\mathbf{q}_{aw} + \mathbf{q}_{bw}\mathbf{q}_{ax} - \mathbf{q}_{bz}\mathbf{q}_{ay} + \mathbf{q}_{by}\mathbf{q}_{az}\right) + \\ \boldsymbol{\omega}_{by}\left(-\mathbf{q}_{by}\mathbf{q}_{aw} + \mathbf{q}_{bz}\mathbf{q}_{ax} + \mathbf{q}_{bw}\mathbf{q}_{ay} - \mathbf{q}_{bx}\mathbf{q}_{az}\right) + \\ \boldsymbol{\omega}_{bz}\left(-\mathbf{q}_{bz}\mathbf{q}_{aw} - \mathbf{q}_{by}\mathbf{q}_{ax} + \mathbf{q}_{bx}\mathbf{q}_{ay} + \mathbf{q}_{bw}\mathbf{q}_{az}\right) \end{matrix}$$

Now we can write $\mathbf{J}$:

$$\mathbf{J}\mathbf{v} = \begin{bmatrix} -\mathbf{q}_{ax}\mathbf{q}_{bw} + \mathbf{q}_{aw}\mathbf{q}_{bx} - \mathbf{q}_{az}\mathbf{q}_{by} + \mathbf{q}_{ay}\mathbf{q}_{bz} \\ -\mathbf{q}_{ay}\mathbf{q}_{bw} + \mathbf{q}_{az}\mathbf{q}_{bx} + \mathbf{q}_{aw}\mathbf{q}_{by} - \mathbf{q}_{ax}\mathbf{q}_{bz} \\ -\mathbf{q}_{az}\mathbf{q}_{bw} - \mathbf{q}_{ay}\mathbf{q}_{bx} + \mathbf{q}_{ax}\mathbf{q}_{by} + \mathbf{q}_{aw}\mathbf{q}_{bz} \\ -\mathbf{q}_{bx}\mathbf{q}_{aw} + \mathbf{q}_{bw}\mathbf{q}_{ax} - \mathbf{q}_{bz}\mathbf{q}_{ay} + \mathbf{q}_{by}\mathbf{q}_{az} \\ -\mathbf{q}_{by}\mathbf{q}_{aw} + \mathbf{q}_{bz}\mathbf{q}_{ax} + \mathbf{q}_{bw}\mathbf{q}_{ay} - \mathbf{q}_{bx}\mathbf{q}_{az} \\ -\mathbf{q}_{bz}\mathbf{q}_{aw} - \mathbf{q}_{by}\mathbf{q}_{ax} + \mathbf{q}_{bx}\mathbf{q}_{ay} + \mathbf{q}_{bw}\mathbf{q}_{az} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \boldsymbol{\omega}_{ax} \\ \boldsymbol{\omega}_{ay} \\ \boldsymbol{\omega}_{az} \\ \boldsymbol{\omega}_{bx} \\ \boldsymbol{\omega}_{by} \\ \boldsymbol{\omega}_{bz} \end{bmatrix}$$

We can also write $\mathbf{J}$ using the quaternion product:

$$\mathbf{J} = \begin{bmatrix} (\mathbf{q}_b\overline{\mathbf{q}}_a)_x \\ (\mathbf{q}_b\overline{\mathbf{q}}_a)_y \\ (\mathbf{q}_b\overline{\mathbf{q}}_a)_z \\ (\mathbf{q}_a\overline{\mathbf{q}}_b)_x \\ (\mathbf{q}_a\overline{\mathbf{q}}_b)_y \\ (\mathbf{q}_a\overline{\mathbf{q}}_b)_z \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} (\mathbf{q}_b\overline{\mathbf{q}}_a)_v \\ (\mathbf{q}_a\overline{\mathbf{q}}_b)_v \end{bmatrix}^{\mathsf{T}} \tag{3.61}$$

A Math Reference

This appendix contains useful definitions, derivations and properties referenced in this document.

A.1 Derivatives

A.1.1 Derivative of a dot product

Here we show the derivative of the dot product between two time dependent vectors:

$$\begin{aligned}
\mathbf{a}(t) \cdot \mathbf{b}(t) &= \mathbf{a}(t)_x \mathbf{b}(t)_x + \mathbf{a}(t)_y \mathbf{b}(t)_y \\
\frac{d}{dt}\left(\mathbf{a}(t) \cdot \mathbf{b}(t)\right) &= \frac{d}{dt}\left(\mathbf{a}(t)_x \mathbf{b}(t)_x\right) + \frac{d}{dt}\left(\mathbf{a}(t)_y \mathbf{b}(t)_y\right) \\
\frac{d}{dt}\left(\mathbf{a}(t)_x \mathbf{b}(t)_x\right) &= \dot{\mathbf{a}}(t)_x \mathbf{b}(t)_x + \mathbf{a}(t)_x \dot{\mathbf{b}}(t)_x \\
\frac{d}{dt}\left(\mathbf{a}(t) \cdot \mathbf{b}(t)\right) &= \dot{\mathbf{a}}(t) \cdot \mathbf{b}(t) + \mathbf{a}(t) \cdot \dot{\mathbf{b}}(t) \qquad \text{(A.1)}
\end{aligned}$$

This can be proven similarly for all $\mathbb{R}^n$ vectors.

A.1.2 Derivative of the length of a vector

Here we show the derivative of the length of a time dependent vector:

$$\begin{aligned}
|\mathbf{a}(t)| &= \sqrt{\mathbf{a}(t) \cdot \mathbf{a}(t)} \\
\frac{d}{dt}|\mathbf{a}(t)| &= \frac{d}{dt}\left(\left(\mathbf{a}(t) \cdot \mathbf{a}(t)\right)^{\frac{1}{2}}\right) \\
&= \frac{1}{2}\left(\mathbf{a}(t) \cdot \mathbf{a}(t)\right)^{-\frac{1}{2}}\left(\mathbf{a}(t) \cdot \dot{\mathbf{a}}(t) + \dot{\mathbf{a}}(t) \cdot \mathbf{a}(t)\right) \\
&= \frac{2\mathbf{a}(t) \cdot \dot{\mathbf{a}}(t)}{2\left(\mathbf{a}(t) \cdot \mathbf{a}(t)\right)^{\frac{1}{2}}} \\
&= \frac{\mathbf{a}(t) \cdot \dot{\mathbf{a}}(t)}{|\mathbf{a}(t)|} \\
&= \frac{\mathbf{a}(t)}{|\mathbf{a}(t)|} \cdot \dot{\mathbf{a}}(t) \qquad \text{(A.2)}
\end{aligned}$$

A.1.3 Derivative of a quaternion

Here we include the derivative of a quaternion for reference[1]:

$$\frac{d}{dt}\mathbf{q} = \left[0, \frac{1}{2}\boldsymbol{\omega}\right]\mathbf{q} \tag{A.3}$$

Where $\boldsymbol{\omega}$ is the angular velocity of $\mathbf{q}$ and is assumed to be constant.

A.1.4 Derivative of a quaternion product

Here we show the derivative of the product of two rotation quaternions. We assume that the angular velocity is constant.

The derivatives of the individual quaternions:

$$\begin{aligned}\frac{d}{dt}\mathbf{o} &= [0, \boldsymbol{\psi}]\,\mathbf{o}\\ \frac{d}{dt}\mathbf{q} &= [0, \boldsymbol{\omega}]\,\mathbf{q}\end{aligned} \tag{A.4}$$

Where $\boldsymbol{\psi}$ is half the angular velocity of $\mathbf{o}$, and $\boldsymbol{\omega}$ is half the angular velocity of $\mathbf{q}$.

We will show the derivative per component. The real part is as follows:

$$\begin{aligned}\frac{d}{dt}(\mathbf{oq})_w &= \frac{d}{dt}\left(\mathbf{o}_w\mathbf{q}_w - \mathbf{o}_x\mathbf{q}_x - \mathbf{o}_y\mathbf{q}_y - \mathbf{o}_z\mathbf{q}_z\right)\\
&= \begin{array}{l}\frac{d}{dt}(\mathbf{o}_w)\,\mathbf{q}_w + \mathbf{o}_w\frac{d}{dt}(\mathbf{q}_w)\\ -\frac{d}{dt}(\mathbf{o}_x)\,\mathbf{q}_x - \mathbf{o}_x\frac{d}{dt}(\mathbf{q}_x)\\ -\frac{d}{dt}(\mathbf{o}_y)\,\mathbf{q}_y - \mathbf{o}_y\frac{d}{dt}(\mathbf{q}_y)\\ -\frac{d}{dt}(\mathbf{o}_z)\,\mathbf{q}_z - \mathbf{o}_z\frac{d}{dt}(\mathbf{q}_z)\end{array}\\
&= \begin{array}{l}([0,\boldsymbol{\psi}]\,\mathbf{o})_w\,\mathbf{q}_w + \mathbf{o}_w\,([0,\boldsymbol{\omega}]\,\mathbf{q})_w\\ -([0,\boldsymbol{\psi}]\,\mathbf{o})_x\,\mathbf{q}_x - \mathbf{o}_x\,([0,\boldsymbol{\omega}]\,\mathbf{q})_x\\ -([0,\boldsymbol{\psi}]\,\mathbf{o})_y\,\mathbf{q}_y - \mathbf{o}_y\,([0,\boldsymbol{\omega}]\,\mathbf{q})_y\\ -([0,\boldsymbol{\psi}]\,\mathbf{o})_z\,\mathbf{q}_z - \mathbf{o}_z\,([0,\boldsymbol{\omega}]\,\mathbf{q})_z\end{array}\\
&= \left(([0,\boldsymbol{\psi}]\,\mathbf{o})\,\mathbf{q} + \mathbf{o}\,([0,\boldsymbol{\omega}]\,\mathbf{q})\right)_w\end{aligned}$$

The other components can be obtained similarly, giving us the same result as with the product rule:

$$\frac{d}{dt}(\mathbf{oq}) = \left(([0,\boldsymbol{\psi}]\,\mathbf{o})\,\mathbf{q} + \mathbf{o}\,([0,\boldsymbol{\omega}]\,\mathbf{q})\right) \tag{A.5}$$

A.2 Vector Identities

A.2.1 Twist angle from vector

We can calculate the twist angle θ of $\mathbf{v}$ about $\mathbf{y}$ as follows (see Figure A.1):

$$\theta = \operatorname{atan2}(\mathbf{z}\cdot\mathbf{v},\ \mathbf{x}\cdot\mathbf{v}) \tag{A.6}$$

Where $\mathbf{x}$ is the reference direction so that $\theta = 0$ when $\mathbf{v}$ lies in the $\mathbf{xy}$ plane.

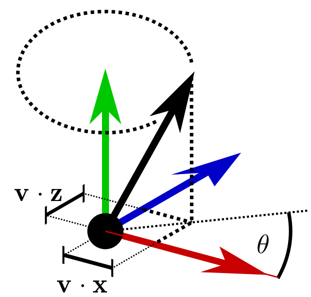


Figure A.1: The twist angle θ of vector $\mathbf{v}$ (black) about twist angle $\mathbf{y}$ from basis $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ (red, green, blue).

A.2.2 Matrix multiplication using dot products

Matrix vector multiplication can be written using dot products:

$$\mathbf{M}\mathbf{v} = \begin{bmatrix} \mathbf{a}_x & \mathbf{a}_y & \mathbf{a}_z \\ \mathbf{b}_x & \mathbf{b}_y & \mathbf{b}_z \\ \mathbf{c}_x & \mathbf{c}_y & \mathbf{c}_z \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \\ \mathbf{v}_z \end{bmatrix} = \begin{bmatrix} \mathbf{a}\cdot\mathbf{v} \\ \mathbf{b}\cdot\mathbf{v} \\ \mathbf{c}\cdot\mathbf{v} \end{bmatrix} \tag{A.7}$$

A.2.3 Cross product matrix

We can write the cross product using a matrix as follows:

$$\begin{aligned} \mathbf{a}\times\mathbf{b} &= [\mathbf{a}]_\times \mathbf{b} \\ [a]_\times &= \begin{bmatrix} 0 & -\mathbf{a}_z & \mathbf{a}_y \\ \mathbf{a}_z & 0 & -\mathbf{a}_x \\ -\mathbf{a}_y & \mathbf{a}_x & 0 \end{bmatrix} \end{aligned} \tag{A.8}$$

Note that $[\mathbf{a}]_\times$ is a skew-symmetric matrix and that the following property holds:

$$\begin{aligned} [\mathbf{a}]_\times^\mathsf{T} &= -[\mathbf{a}]_\times \\ -\mathbf{a}\times\mathbf{b} = \mathbf{b}\times\mathbf{a} &= -[\mathbf{a}]_\times \mathbf{b} = [\mathbf{a}]_\times^\mathsf{T} \mathbf{b} \end{aligned} \tag{A.9}$$

A.3 Quaternion Identities

A.3.1 Quaternion axis angle formula

Here we include the formula to construct a quaternion that represents a rotation of a radians about (unit length) axis $\mathbf{b}$ for reference.

$$\mathbf{q} = \left[\cos\left(\frac{1}{2}a\right), \sin\left(\frac{1}{2}a\right)\mathbf{b}\right] \tag{A.10}$$

We define a function $\mathrm{q}\odot(\mathbf{b}, a) = \mathbf{q}$ for convenience.

A.3.2 Quaternion product using scalar and vector parts

We can write the quaternion product using scalar and vector operations:

$$\mathbf{ab} = [(\mathbf{a}_w\mathbf{b}_w - \mathbf{a}_i \cdot \mathbf{b}_i)\,, \quad (\mathbf{a}_w\mathbf{b}_i + \mathbf{b}_w\mathbf{a}_i + \mathbf{a}_i \times \mathbf{b}_i)] \tag{A.11}$$

Where $\mathbf{q}_i$ is the imaginary vector of $\mathbf{q}$, and $\mathbf{q}_w$ is the real part of $\mathbf{q}$.

A.3.3 Relative angle

By the definition of multiplication for quaternions $\mathbf{a}$ and $\mathbf{b}$:

$$\mathbf{ab} \cdot \mathbf{w} = \mathbf{a}_w\mathbf{b}_w - \mathbf{a}_{xyz} \cdot \mathbf{b}_{xyz} = (\mathbf{ab})_w \tag{A.12}$$

Where $\mathbf{w} = [1, 0, 0, 0]$ and where $\cdot$ is the common vector dot product.

For rotation/unit length quaternions $\mathbf{a}$ and $\mathbf{b}$:

$$\mathbf{a}^{-1} = \overline{\mathbf{a}} = [\mathbf{a}_w, -\mathbf{a}_{xyz}] \tag{A.13}$$

Which gives:

$$\begin{aligned} \mathbf{ab}^{-1} \cdot \mathbf{w} &= \mathbf{a}_w\mathbf{b}_w + \mathbf{a}_{xyz} \cdot \mathbf{b}_{xyz} \\ &= \mathbf{a} \cdot \mathbf{b} \end{aligned} \tag{A.14}$$

This can be used to determine the angle between two quaternions:

$$\angle(\mathbf{a}, \mathbf{b}) = 2\,\mathrm{acos}(\mathbf{a} \cdot \mathbf{b}) \tag{A.15}$$

A.3.4 Quaternion product using matrices

We can write the quaternion product $\mathbf{ab}$ using a 4×4 matrix vector multiplication as follows[9]:

$$\begin{aligned} \mathbf{ab} &= \begin{bmatrix} \mathbf{a}_w\mathbf{b}_w - \mathbf{a}_x\mathbf{b}_x - \mathbf{a}_y\mathbf{b}_y - \mathbf{a}_z\mathbf{b}_z \\ \mathbf{a}_w\mathbf{b}_x + \mathbf{a}_x\mathbf{b}_w + \mathbf{a}_y\mathbf{b}_z - \mathbf{a}_z\mathbf{b}_y \\ \mathbf{a}_w\mathbf{b}_y - \mathbf{a}_x\mathbf{b}_z + \mathbf{a}_y\mathbf{b}_w + \mathbf{a}_z\mathbf{b}_x \\ \mathbf{a}_w\mathbf{b}_z + \mathbf{a}_x\mathbf{b}_y - \mathbf{a}_y\mathbf{b}_x + \mathbf{a}_z\mathbf{b}_w \end{bmatrix} \\ \mathrm{M_L}(\mathbf{a})\mathbf{b} &= \mathbf{Ab} = \mathbf{ab} \\ \mathbf{Ab} &= \begin{bmatrix} \mathbf{a}_w & -\mathbf{a}_x & -\mathbf{a}_y & -\mathbf{a}_z \\ \mathbf{a}_x & \mathbf{a}_w & -\mathbf{a}_z & \mathbf{a}_y \\ \mathbf{a}_y & \mathbf{a}_z & \mathbf{a}_w & -\mathbf{a}_x \\ \mathbf{a}_z & -\mathbf{a}_y & \mathbf{a}_x & \mathbf{a}_w \end{bmatrix} \begin{bmatrix} \mathbf{b}_w \\ \mathbf{b}_x \\ \mathbf{b}_y \\ \mathbf{b}_z \end{bmatrix} \end{aligned} \tag{A.16}$$

Similarly with $\mathbf{b}$ as a matrix:

$$\begin{aligned} \mathrm{M_R}(\mathbf{b})\mathbf{a} &= \mathbf{Ba} = \mathbf{ab} \\ \mathbf{Ba} &= \begin{bmatrix} \mathbf{b}_w & -\mathbf{b}_x & -\mathbf{b}_y & -\mathbf{b}_z \\ \mathbf{b}_x & \mathbf{b}_w & \mathbf{b}_z & -\mathbf{b}_y \\ \mathbf{b}_y & -\mathbf{b}_z & \mathbf{b}_w & \mathbf{b}_x \\ \mathbf{b}_z & \mathbf{b}_y & -\mathbf{b}_x & \mathbf{b}_w \end{bmatrix} \begin{bmatrix} \mathbf{a}_w \\ \mathbf{a}_x \\ \mathbf{a}_y \\ \mathbf{a}_z \end{bmatrix} \end{aligned} \tag{A.17}$$

Note that conjugation can be related to the matrix transpose:

$$\begin{aligned} \mathrm{M_L}(\mathbf{a})^\intercal &= \mathrm{M_L}(\overline{\mathbf{a}}) \\ \mathrm{M_R}(\mathbf{b})^\intercal &= \mathrm{M_R}(\overline{\mathbf{b}}) \end{aligned} \tag{A.18}$$

Also note that the matrix is skew symmetric for pure imaginary quaternions:

$$\left.\begin{aligned} \mathrm{M_L}(\mathbf{a})^\intercal &= -\,\mathrm{M_L}(\mathbf{a}) \\ \mathrm{M_R}(\mathbf{a})^\intercal &= -\,\mathrm{M_R}(\mathbf{a}) \end{aligned}\right\}\text{for } \mathbf{a}_w = 0 \tag{A.19}$$

A.3.5 Rotation quaternion inverse factor reverse

It is possible to prove some quaternion properties similar to matrices as we can represent a rotation quaternion with a 3×3 matrix. The inverse of a rotation matrix is its transpose:

$$\begin{aligned} \mathbf{A} &= \mathrm{R}(\mathbf{q}) \\ \mathbf{q}^{-1} &= \overline{\mathbf{q}} \\ \mathbf{A}^{-1} &= \mathbf{A}^\intercal \\ \mathrm{R}(\mathbf{q}^{-1}) &= \mathrm{R}(\overline{\mathbf{q}}) = \mathbf{A}^\intercal \end{aligned} \tag{A.20}$$

Where $\mathrm{R}(\mathbf{q})$ is the rotation matrix constructed from $\mathbf{q}$.

This makes it possible to reverse the factors of an inverted quaternion multiplication:

$$\begin{aligned} \mathbf{B} &= \mathrm{R}(\mathbf{r}) \\ \mathrm{R}\left((\mathbf{qr})^{-1}\right) &= (\mathbf{AB})^\intercal = \mathbf{B}^\intercal\mathbf{A}^\intercal = \mathrm{R}(\mathbf{r}^{-1})\,\mathrm{R}(\mathbf{q}^{-1}) \\ (\mathbf{qr})^{-1} &= \mathbf{r}^{-1}\mathbf{q}^{-1} \end{aligned} \tag{A.21}$$

A.3.6 Shortest rotation between two directions

We can create a quaternion $\mathbf{q}_{ab}$ that describes the shortest rotation from vector $\mathbf{a}$ to vector $\mathbf{b}$ as follows:

$$\mathbf{q}_{ab} = [\mathbf{a} \cdot \mathbf{h}, \mathbf{a} \times \mathbf{h}] \tag{A.22}$$

With

$$\mathbf{h} = \begin{cases} \frac{\mathbf{a}+\mathbf{b}}{|\mathbf{a}+\mathbf{b}|}, & \text{if } |\mathbf{a}+\mathbf{b}| \neq 0 \\ \text{any tangent of } \mathbf{a}, & \text{otherwise} \end{cases} \tag{A.23}$$

Note that $\mathbf{q}_{ab}$ becomes an identity quaternion if $\mathbf{a} = \mathbf{b}$.

A.3.7 Swing twist decomposition

We can split quaternion $\mathbf{q}$ in two parts so that:

$$\mathbf{q} = \mathbf{q}_{\text{twist}}\mathbf{q}_{\text{swing}} \tag{A.24}$$

where $\mathbf{q}_{\text{twist}}$ only rotates about twist axis $\mathbf{v}$. We know that the rotation axis of $\mathbf{q}_{\text{twist}}$ must be parallel with $\mathbf{v}$. We can calculate $\mathbf{q}_{\text{twist}}$ from $\mathbf{q}$ by projecting its rotation axis on $\mathbf{v}$:

$$\mathbf{q}_{\text{twist}} = \frac{[\mathbf{q}_w, \mathbf{q}_\mathbf{i} \cdot \mathbf{v}]}{|[\mathbf{q}_w, \mathbf{q}_\mathbf{i} \cdot \mathbf{v}]|} \tag{A.25}$$

Normalization is required to keep a unit quaternion.

Knowing $\mathbf{q}_{\text{twist}}$ we can solve (A.24) for $\mathbf{q}_{\text{swing}}$:

$$\overline{\mathbf{q}}_{\text{twist}}\mathbf{q} = \mathbf{q}_{\text{swing}} \tag{A.26}$$

Note that the following equality holds:

$$2 \operatorname{acos}\left(\mathbf{q}_{\text{swing}w}\right) = \operatorname{acos}\left((\mathbf{q}\mathbf{v}\overline{\mathbf{q}}) \cdot \mathbf{v}\right) \tag{A.27}$$

B Additional Derivations

We include some alternate and additional derivations in this appendix for reference.

B.1 Derivation of force solution

Here we show how to derive the same equation as in Section 2.1, but using forces instead of impulses. This derivation is included as reference because most other works use λ as the signed magnitude of the force, instead of the signed magnitude of the impulse.

We start with the following system of equations. Euler's method:

$$\begin{aligned} \mathbf{v} &= \overline{\mathbf{v}} + \Delta t \mathbf{M}^{-1}\mathbf{f} \\ \mathbf{M}\frac{\mathbf{v}-\overline{\mathbf{v}}}{\Delta t} &= \mathbf{f} \end{aligned} \tag{B.1}$$

where Δt is the time step.

Newton's second law:

$$\mathbf{f} = \mathbf{M}\mathbf{a} = \mathbf{M}\frac{\mathbf{v}-\overline{\mathbf{v}}}{\Delta t} \tag{B.2}$$

Virtual work:

$$\mathbf{f} = \mathbf{J}^{\mathsf{T}}\boldsymbol{\lambda} \tag{B.3}$$

Velocity constraint:

$$0 = \mathbf{J}\mathbf{v} + b \tag{B.4}$$

Combining the above equations lets us solve for λ:

$$\begin{aligned} 0 &= \mathbf{J}\left(\mathbf{M}^{-1}\mathbf{f}\Delta t + \overline{\mathbf{v}}\right) + b \\ &= \mathbf{J}\mathbf{M}^{-1}\mathbf{f}\Delta t + \mathbf{J}\overline{\mathbf{v}} + b \\ &= \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\mathsf{T}}\boldsymbol{\lambda}\Delta t + \mathbf{J}\overline{\mathbf{v}} + b \\ -\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\mathsf{T}}\boldsymbol{\lambda}\Delta t &= \mathbf{J}\overline{\mathbf{v}} + b \\ \boldsymbol{\lambda}\Delta t &= \frac{\mathbf{J}\overline{\mathbf{v}} + b}{-\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\mathsf{T}}} \end{aligned} \tag{B.5}$$

This results in the same equation as (2.4) given that $\mathbf{p} = \mathbf{f}\Delta t$.

B.2 Derivation of the impulse-based reaction model

Here we show that (2.4) is closely related to the impulse based reaction model (also known to some as the Eberly formula). And in the case where $\mathbf{j}$ is the constraint Jacobian of a contact constraint (see Section 3.1.2, or slide 21 of [5]), they are the same if the restitution coefficient $\epsilon = 0$ and $b = 0$. This solution is also discussed in [11].

We start with Newton's laws of motion for the linear velocity of bodies a and b:

$$\mathbf{v}_a = \overline{\mathbf{v}}_a - m_a^{-1} p_c \mathbf{n} \tag{B.6a}$$

$$\mathbf{v}_b = \overline{\mathbf{v}}_b + m_b^{-1} p_c \mathbf{n} \tag{B.6b}$$

And for the angular velocities:

$$\boldsymbol{\omega}_a = \overline{\boldsymbol{\omega}}_a - p_c \mathbf{I}_a^{-1} (\mathbf{r}_a \times \mathbf{n}) \tag{B.7a}$$

$$\boldsymbol{\omega}_b = \overline{\boldsymbol{\omega}}_b + p_c \mathbf{I}_b^{-1} (\mathbf{r}_b \times \mathbf{n}) \tag{B.7b}$$

Where $\overline{\mathbf{v}}_i$ and $\overline{\boldsymbol{\omega}}_i$ are the pre-collision velocities of body i, $\mathbf{v}_i$ and $\boldsymbol{\omega}_i$ are the post-collision velocities, p_c is the reaction impulse, $\mathbf{r}_i$ is contact offset relative to the center of mass of body i, $\mathbf{n}$ is the contact normal, and $\mathbf{I}_i$ is the inertia tensor of body i. Note the similarities with (2.1) where the linear and rotational parts are combined.

We calculate the contact point velocity $\mathbf{v}_{pi}$ of body i as follows:

$$\mathbf{v}_{pi} = \mathbf{v}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i \tag{B.8}$$

And similar for the pre-resolution velocity $\overline{\mathbf{v}}_{pi}$.

We calculate the relative contact point velocity $\mathbf{v}_r$ as follows:

$$\mathbf{v}_r = \mathbf{v}_{pb} - \mathbf{v}_{pa} \tag{B.9}$$

We know the relationship between the pre and post-collision contact point velocity:

$$\mathbf{v}_r \cdot \mathbf{n} = -\epsilon \overline{\mathbf{v}}_r \cdot \mathbf{n} \tag{B.10}$$

Where $\boldsymbol{\epsilon}$ is the coefficient of restitution. This is a similar relationship between $\overline{\mathbf{v}}$ and $\mathbf{v}$ as (2.3) describes.

Before we start solving for $\mathbf{p}_c$ we will expand (B.10). First we substitute (B.9) into (B.10):

$$(\mathbf{v}_{pb} - \mathbf{v}_{pa}) \cdot \mathbf{n} = -\epsilon (\overline{\mathbf{v}}_{pb} - \overline{\mathbf{v}}_{pa}) \cdot \mathbf{n} \tag{B.11}$$

Now we can substitute (B.8):

$$\begin{gathered}(\mathbf{v}_b + \boldsymbol{\omega}_b \times \mathbf{r}_b) \cdot \mathbf{n} - (\mathbf{v}_a + \boldsymbol{\omega}_a \times \mathbf{r}_a) \cdot \mathbf{n} = \\ -\epsilon (\overline{\mathbf{v}}_b + \overline{\boldsymbol{\omega}}_b \times \mathbf{r}_b) \cdot \mathbf{n} + \epsilon (\overline{\mathbf{v}}_a + \overline{\boldsymbol{\omega}}_a \times \mathbf{r}_a) \cdot \mathbf{n}\end{gathered}$$

And finally we substitute (B.6):

$$\left(\left(\overline{\mathbf{v}}_b + m_b^{-1} p_c \mathbf{n}\right) + \boldsymbol{\omega}_b \times \mathbf{r}_b\right) \cdot \mathbf{n} - \left(\left(\overline{\mathbf{v}}_a - m_a^{-1} p_c \mathbf{n}\right) + \boldsymbol{\omega}_a \times \mathbf{r}_a\right) \cdot \mathbf{n} = -\epsilon \left(\overline{\mathbf{v}}_{pb} - \overline{\mathbf{v}}_{pa}\right) \cdot \mathbf{n}$$

And (B.7):

$$\begin{aligned} &\left(\left(\overline{\mathbf{v}}_b + m_b^{-1} p_c \mathbf{n}\right) + \boldsymbol{\omega}_b \times \mathbf{r}_b\right) \cdot \mathbf{n} \\ &- \left(\left(\overline{\mathbf{v}}_a - m_a^{-1} p_c \mathbf{n}\right) + \boldsymbol{\omega}_a \times \mathbf{r}_a\right) \cdot \mathbf{n} = \\ &-\epsilon \left(\overline{\mathbf{v}}_{pb} - \overline{\mathbf{v}}_{pa}\right) \cdot \mathbf{n} \end{aligned}$$

$$\begin{aligned} &\left(\left(\overline{\mathbf{v}}_b + m_b^{-1} p_c \mathbf{n}\right) + \left(\overline{\boldsymbol{\omega}}_b + p_c \mathbf{I}_b^{-1} \left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n} \\ &- \left(\left(\overline{\mathbf{v}}_a - m_a^{-1} p_c \mathbf{n}\right) + \left(\overline{\boldsymbol{\omega}}_a - p_c \mathbf{I}_a^{-1} \left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n} = \\ &-\epsilon \left(\overline{\mathbf{v}}_{pb} - \overline{\mathbf{v}}_{pa}\right) \cdot \mathbf{n} \end{aligned}$$

We will now isolate p_c, starting with the first part:

$$\begin{aligned} &\left(\left(\overline{\mathbf{v}}_b + m_b^{-1} p_c \mathbf{n}\right) + \left(\overline{\boldsymbol{\omega}}_b + p_c \mathbf{I}_b^{-1} \left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n} \ldots \\ &\left(\overline{\mathbf{v}}_b + m_b^{-1} p_c \mathbf{n}\right) \cdot \mathbf{n} + \left(\left(\overline{\boldsymbol{\omega}}_b + p_c \mathbf{I}_b^{-1} \left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n} \ldots \\ &\overline{\mathbf{v}}_b \mathbf{n} + m_b^{-1} p_c \mathbf{n} \cdot \mathbf{n} + \left(\left(\overline{\boldsymbol{\omega}}_b + p_c \mathbf{I}_b^{-1} \left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n} \ldots \\ &\overline{\mathbf{v}}_b \mathbf{n} + m_b^{-1} p_c \mathbf{n} \cdot \mathbf{n} + \left(\overline{\boldsymbol{\omega}}_b \times \mathbf{r}_b + p_c \mathbf{I}_b^{-1} \left(\mathbf{r}_b \times \mathbf{n}\right) \times \mathbf{r}_b\right) \cdot \mathbf{n} \ldots \\ &\overline{\mathbf{v}}_b \mathbf{n} + m_b^{-1} p_c \mathbf{n} \cdot \mathbf{n} + \left(\overline{\boldsymbol{\omega}}_b \times \mathbf{r}_b\right) \cdot \mathbf{n} + p_c \left(\left(\mathbf{I}_b^{-1} \left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n} \ldots \\ &\overline{\mathbf{v}}_b \mathbf{n} + \left(m_b^{-1} \mathbf{n} \cdot \mathbf{n}\right) p_c + \left(\overline{\boldsymbol{\omega}}_b \times \mathbf{r}_b\right) \cdot \mathbf{n} + p_c \left(\left(\mathbf{I}_b^{-1} \left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n} \ldots \\ &\overline{\mathbf{v}}_b \mathbf{n} + \left(\overline{\boldsymbol{\omega}}_b \times \mathbf{r}_b\right) \cdot \mathbf{n} + \left(m_b^{-1} \mathbf{n} \cdot \mathbf{n}\right) p_c + p_c \left(\left(\mathbf{I}_b^{-1} \left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n} \ldots \\ &\overline{\mathbf{v}}_b \mathbf{n} + \left(\overline{\boldsymbol{\omega}}_b \times \mathbf{r}_b\right) \cdot \mathbf{n} + p_c \left(m_b^{-1} \mathbf{n} \cdot \mathbf{n} + \left(\left(\mathbf{I}_b^{-1} \left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n}\right) \ldots \end{aligned}$$

We substitute Equation (B.8) again:

$$\begin{aligned} &\overline{\mathbf{v}}_b \cdot \mathbf{n} + \left(\overline{\boldsymbol{\omega}}_b \times \mathbf{r}_b\right) \cdot \mathbf{n} + p_c \left(m_b^{-1} \mathbf{n} \cdot \mathbf{n} + \left(\left(\mathbf{I}_b^{-1} \left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n}\right) \ldots \\ &\overline{\mathbf{v}}_{pb} \cdot \mathbf{n} + p_c \left(m_b^{-1} \mathbf{n} \cdot \mathbf{n} + \left(\left(\mathbf{I}_b^{-1} \left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n}\right) \ldots \end{aligned}$$

$\mathbf{n} \cdot \mathbf{n} = 1$ as it is a unit vector:

$$\overline{\mathbf{v}}_{pb} \cdot \mathbf{n} + p_c \left(m_b^{-1} + \left(\left(\mathbf{I}_b^{-1} \left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n}\right) \ldots \tag{B.12}$$

And similar with the second part:

$$\begin{aligned} &\ldots - \left(\left(\overline{\mathbf{v}}_a - m_a^{-1} p_c \mathbf{n}\right) + \left(\overline{\boldsymbol{\omega}}_a - p_c \mathbf{I}_a^{-1} \left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n} \\ &\ldots - \left(\overline{\mathbf{v}}_a - m_a^{-1} p_c \mathbf{n}\right) \cdot \mathbf{n} - \left(\left(\overline{\boldsymbol{\omega}}_a - p_c \mathbf{I}_a^{-1} \left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n} \\ &\ldots - \left(\overline{\mathbf{v}}_a - m_a^{-1} p_c \mathbf{n}\right) \cdot \mathbf{n} - \left(\overline{\boldsymbol{\omega}}_a \times \mathbf{r}_a - p_c \left(\mathbf{I}_a^{-1} \left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n} \\ &\ldots - \overline{\mathbf{v}}_a \cdot \mathbf{n} + m_a^{-1} p_c \mathbf{n} \cdot \mathbf{n} - \left(\overline{\boldsymbol{\omega}}_a \times \mathbf{r}_a\right) \cdot \mathbf{n} + p_c \left(\left(\mathbf{I}_a^{-1} \left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n} \\ &\ldots - \overline{\mathbf{v}}_a \cdot \mathbf{n} - \left(\overline{\boldsymbol{\omega}}_a \times \mathbf{r}_a\right) \cdot \mathbf{n} + p_c \left(m_a^{-1} + \left(\left(\mathbf{I}_a^{-1} \left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n}\right) \\ &\ldots - \overline{\mathbf{v}}_{pa} \cdot \mathbf{n} + p_c \left(m_a^{-1} + \left(\left(\mathbf{I}_a^{-1} \left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n}\right) \end{aligned}$$

Putting the parts back together we can solve for p_c:

$$\begin{aligned}
\overline{\mathbf{v}}_{pb} \cdot \mathbf{n} + p_c \left(m_b^{-1} + \left(\left(\mathbf{I}_a^{-1}\left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n}\right) \\
-\overline{\mathbf{v}}_{pa} \cdot \mathbf{n} + p_c \left(m_a^{-1} + \left(\left(\mathbf{I}_a^{-1}\left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n}\right) = \\
-\epsilon \left(\overline{\mathbf{v}}_{pb} - \overline{\mathbf{v}}_{pa}\right) \cdot \mathbf{n}
\end{aligned}$$

$$\begin{aligned}
\left(\overline{\mathbf{v}}_{pb} - \overline{\mathbf{v}}_{pa}\right) \cdot \mathbf{n} + \\
p_c \left(m_b^{-1} + m_a^{-1} + \left(\left(\mathbf{I}_b^{-1}\left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b + \left(\mathbf{I}_a^{-1}\left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n}\right) = \\
-\epsilon \left(\overline{\mathbf{v}}_{pb} - \overline{\mathbf{v}}_{pa}\right) \cdot \mathbf{n}
\end{aligned}$$

$$\begin{aligned}
\left(\overline{\mathbf{v}}_{pb} - \overline{\mathbf{v}}_{pa}\right) \cdot \mathbf{n} + \epsilon \left(\overline{\mathbf{v}}_{pb} - \overline{\mathbf{v}}_{pa}\right) \cdot \mathbf{n} = \\
-p_c \left(m_b^{-1} + m_a^{-1} + \left(\left(\mathbf{I}_b^{-1}\left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b + \left(\mathbf{I}_a^{-1}\left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n}\right)
\end{aligned}$$

$$\begin{aligned}
\left(\epsilon + 1\right) \left(\overline{\mathbf{v}}_{pb} - \overline{\mathbf{v}}_{pa}\right) \cdot \mathbf{n} = \\
-p_c \left(m_b^{-1} + m_a^{-1} + \left(\left(\mathbf{I}_b^{-1}\left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b + \left(\mathbf{I}_a^{-1}\left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n}\right)
\end{aligned}$$

$$\frac{-\left(\epsilon + 1\right) \left(\overline{\mathbf{v}}_{pb} - \overline{\mathbf{v}}_{pa}\right) \cdot \mathbf{n}}{m_b^{-1} + m_a^{-1} + \left(\left(\mathbf{I}_b^{-1}\left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b + \left(\mathbf{I}_a^{-1}\left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n}} = p_c$$

And when we substitute (B.9) we arrive at the impulse reaction:

$$\frac{-\left(\epsilon + 1\right) \left(\overline{\mathbf{v}}_{r}\right) \cdot \mathbf{n}}{m_b^{-1} + m_a^{-1} + \left(\left(\mathbf{I}_b^{-1}\left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b + \left(\mathbf{I}_a^{-1}\left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a\right) \cdot \mathbf{n}} = p_c \tag{B.13}$$

This is the same as (2.4) but instead of a bias velocity b there is a restitution coefficient ϵ.

B.3 Effective Mass Relation

Here we show that the effective mass (the denominator from (2.4) for contact constraints, also shown in slide 55 of [6]), is equivalent to the denominator of (B.13).

We start by filling in (2.6) with $\mathbf{j}$ from Section 3.1.2 and $\mathbf{M}^{-1}$ from (2.5):

$$\begin{aligned}
\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\intercal} = & \left(-\mathbf{n}\right) \cdot \left(-\mathbf{n}\right) m_a^{-1} + \left(\mathbf{I}_a^{-1}\left(-\mathbf{r}_a \times \mathbf{n}\right)\right) \cdot \left(-\mathbf{r}_a \times \mathbf{n}\right) \\
& + \mathbf{n} \cdot \mathbf{n} m_b^{-1} + \left(\mathbf{I}_b^{-1}\left(\mathbf{r}_b \times \mathbf{n}\right)\right) \cdot \left(\mathbf{r}_b \times \mathbf{n}\right) \\
= & \ \mathbf{n} \cdot \mathbf{n} m_a^{-1} + \left(\mathbf{I}_a^{-1}\left(\mathbf{r}_a \times \mathbf{n}\right)\right) \cdot \left(\mathbf{r}_a \times \mathbf{n}\right) \\
& + \mathbf{n} \cdot \mathbf{n} m_b^{-1} + \left(\mathbf{I}_b^{-1}\left(\mathbf{r}_b \times \mathbf{n}\right)\right) \cdot \left(\mathbf{r}_b \times \mathbf{n}\right) \\
= & \ m_a^{-1} + \left(\mathbf{I}_a^{-1}\left(\mathbf{r}_a \times \mathbf{n}\right)\right) \cdot \left(\mathbf{r}_a \times \mathbf{n}\right) + m_b^{-1} + \left(\mathbf{I}_b^{-1}\left(\mathbf{r}_b \times \mathbf{n}\right)\right) \cdot \left(\mathbf{r}_b \times \mathbf{n}\right) \\
= & \ m_a^{-1} + m_b^{-1} + \left(\mathbf{I}_a^{-1}\left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a \cdot \mathbf{n} + \left(\mathbf{I}_b^{-1}\left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b \cdot \mathbf{n} \\
= & \ m_a^{-1} + m_b^{-1} + \left(\left(\mathbf{I}_a^{-1}\left(\mathbf{r}_a \times \mathbf{n}\right)\right) \times \mathbf{r}_a + \left(\mathbf{I}_b^{-1}\left(\mathbf{r}_b \times \mathbf{n}\right)\right) \times \mathbf{r}_b\right) \cdot \mathbf{n}
\end{aligned} \tag{B.14}$$

Acknowledgements

We would like to thank Nixxes Software for providing the opportunity and work space to complete this project. We would like to thank the people at Crystal Dynamics, Harsha Kalangi and Gary Snethen in particular, for their contributions to the constraint equations and numerous insightful discussions. We would like to thank Michiel Roza from Nixxes and Bert Heesakkers from NHTV Breda University of Applied Sciences for proofreading. And we would also like the thank Jurriaan Tamis for providing the cover image.

Bibliography

[1] David Baraff. An introduction to physically based modeling: Rigid body simulation II-nonpenetration constraints. *SIGGRAPH course notes*, pages D31–D68, 1997. URL https://www.cs.cmu.edu/~baraff/sigcourse/notesd2.pdf.

[2] Marco Carricato. Homokinetic transmission of rotational motion via constant-velocity joints in closed-chain wrists. In *12th World Congress in Mechanism and Machine Science, Besançon, France*, pages 284–290, 2007.

[3] E. Catto. Iterative Dynamics with Temporal Coherence. *Game Developer Conference*, January 2005. URL http://www.bulletphysics.com/ftp/pub/test/physics/papers/IterativeDynamics.pdf.

[4] Erin Catto. Baumgarte and softness for 2d joints, 2005. URL http://bulletphysics.org/Bullet/phpBB3/viewtopic.php?f=4&t=1354#p5044. Physics Simulation Forum - Last visited on 21-5-2015.

[5] Erin Catto. Fast and simple physics using sequential impulses. In *Games Developer Conference*, 2006.

[6] Erin Catto. Modeling and solving constraints. In *Games Developer Conference*, 2008.

[7] Erin Catto. Soft constraints. In *Game Developers Conference*, 2011.

[8] Daniel Chappuis. *Constraints Derivation for Rigid Body Simulation in 3D*, 2013. URL http://danielchappuis.ch/download/ConstraintsDerivationRigidBody3D.pdf.

[9] Berthold KP Horn. Some notes on unit quaternions and rotation. *Lecture handouts*, 2001. URL http://www.ocw.nur.ac.rw/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-801Fall-2004/6A9E9F35-52E5-44C7-885A-ABB949051641/0/quaternions.pdf.

[10] Kim Min-Seok, Park Su-Jin, YOO Wan-Suk, and KIM Doo-Hyun. Detc2005-84175 constraint formulation of constant velocity joints to estimate joint reaction torque. *PROCEEDINGS OF THE ASME DESIGN ENGINEERING TECHNICAL CONFERENCES; 6; 2339-2344*, 2005.

[11] Wikipedia. Collision response — Wikipedia, the free encyclopedia, 2014. URL http://en.wikipedia.org/w/index.php?title=Collision_response&oldid=633303309. [Online; accessed 26-March-2015].