

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Dynamics and Control of Multibody Systems

Marek Vondrak¹, Leonid Sigal² and Odest Chadwicke Jenkins¹

¹*Brown University,*

²*University of Toronto*

¹*U.S.A.,*

²*Canada*

1. Introduction

Over the past decade, physics-based simulation has become a key enabling technology for variety of applications. It has taken a front seat role in computer games, animation of virtual worlds and robotic simulation. New applications are still emerging and physics is becoming an integral part of many new technologies that might have been thought of not being directly related to physics. For example, physics has been recently used to explain and recover the motion of the subject from video (Vondrak et al., 2008). Unfortunately, despite the availability of various simulation packages, the level of expertise required to use physical simulation correctly is quite high. The goal of this chapter is thus to establish sufficiently strong grounds that would allow the reader to not only understand and use existing simulation packages properly but also to implement their own solutions if necessary. We choose to model world as a set of constrained rigid bodies as this is the most commonly used approximation to real world physics and such a model is able to deliver predictable high quality results in real time. To make sure bodies, affected by various forces, move as desired, a mechanism for controlling motion through the use of constraints is introduced. We then apply the approach to the problem of physics-based animation (control) of humanoid characters.

We start with a review of unconstrained rigid body dynamics and introduce the basic concepts like body mass properties, state parameterization and equations of motion. The derivations will follow (Baraff et al., 1997) and (Erleben, 2002), using notation from (Baraff, 1996). For background information, we recommend reading (Eberly, 2003; Thornton et al., 2003; Bourg, 2002). We then move to Lagrangian constrained rigid body dynamics and show how constraints on body accelerations, velocities or positions can be modeled and incorporated into simpler unconstrained rigid body dynamics. Various kinds of constraints are discussed, including equality constraints (required for the implementation of “joint motors”), inequality constraints (used for the implementation of “joint angle limits”) and bounded equality constraints (used for implementation of motors capable of generating limited motor forces). We then reduce the problem of solving for constraint forces to the problem of solving linear complementarity problems. Finally, we show how this method can be used to enforce body non-penetration and implement a contact model, (Trinkle et al., 1997; Kawachi et al., 1997).

Source: Motion Control, Book edited by: Federico Casolo,
ISBN 978-953-7619-55-8, pp. 580, January 2010, INTECH, Croatia, downloaded from SCIYO.COM

Lastly, we illustrate how before mentioned constraints can be used to implement composite articulated bodies and how these bodies can be actuated by generating appropriate motor torques at joints, following (Kokkevis, 2004). Various kinds of convenient joint parameterizations with different degrees of freedom, together with options for their actuation, are discussed.

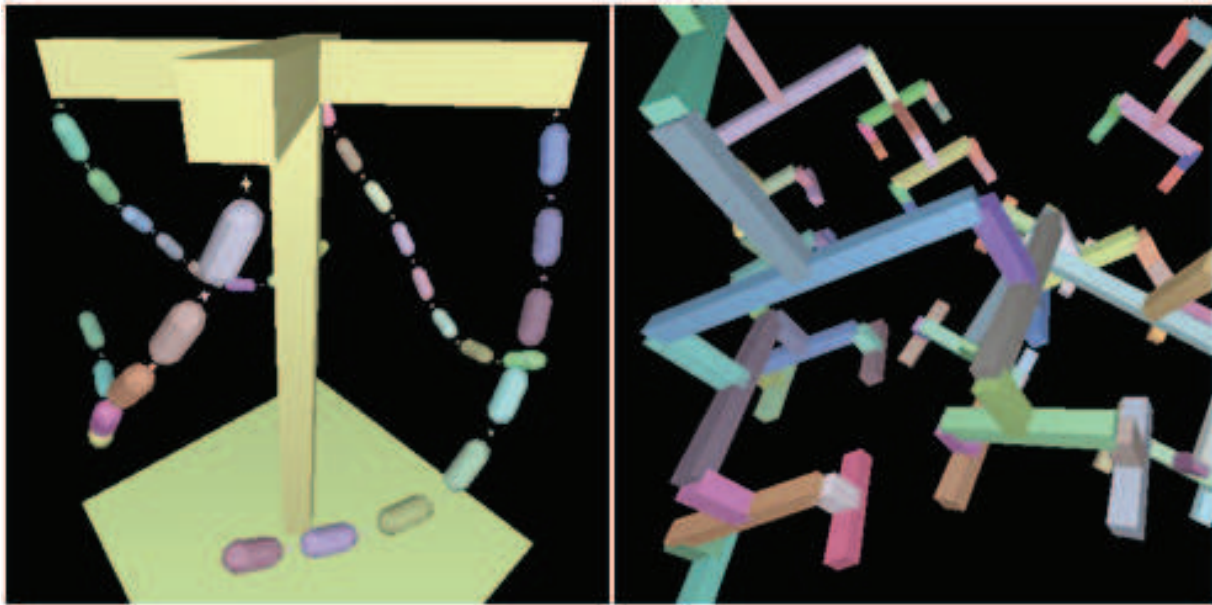


Fig. 1. Examples of constrained rigid body systems. Constraints glue bodies together at designated points, actuate the structures or enforce non-penetration.

1.1 Related work

While physical simulation is conceptually well understood, control of articulated high degree of freedom bodies (or characters) remains a challenging problem. On the simulation side there currently exist a number of commercial and open source engines that deliver robust and computationally efficient performance (e.g., Crisis, Havoc, Newton, Open Dynamics Engine (ODE), PhysX). Quantitative analysis of performance among some of these and other popular choices are discussed in (Boeing et al., 2007). However, control over the motion of characters within these simulators is still very limited. Those packages that do provide means for building user defined dynamic controllers (e.g., Euphoria by NaturalMotion and Dynamic Controller Toolbox (Shapiro et al., 2007)) still lack fidelity and ability to model stylistic variations that are important for producing realistic motions.

In this chapter, we describe trajectory-based control (either in terms of joint angles or rigidly attached points) implemented in the form of constraints. This type of the control is simple, general, stable, and is available (or easy to implement) within any simulator environment that supports constraints (e.g., Crisis, ODE, Newton). That said, other control strategies have also been proposed and are applicable for appropriate domains and tasks. For example, where modeling of high fidelity trajectories is hard, one can resort to sparse set of key-poses with proportional derivative (PD) control (Hodgins et al., 1995); such controllers can produce very stable motions (e.g., human gait (Yin et al., 2007)) but often look artificial or robotic. Locomotion controllers with stable limit cycle behavior are popular and appealing

choices for various forms of cyclic gates (Laszlo et al, 1996); particularly in the robotics and biomechanics communities (Goswami et al., 1996).

At least in part the challenges in control stem from the high dimensionality of the control space. To that end few approaches have attempted to learn low-dimensional controllers through optimization (Safonova et al., 2004). Other optimization-based techniques are also popular, but often require initial motion (Liu et al., 2005) or existing controller (Yin et al., 2008) for adaptation to new environmental conditions or execution speed (McCann et al., 2006). Furthermore, because it is unlikely that a single controller can produce complex motions of interest, approaches that focus on building composable controllers (Faloutsos et al., 2001) have also been explored. Alternatively, controllers that attempt to control high degree-of-freedom motions using task-based formulations, that allow decoupling and composing of controls required to complete a particular task (e.g., maintain balance) from controls required to actuate redundant degrees of freedom with respect to the task, are also appealing (Abe et al., 2006). In robotics such strategies are known as operational space control (Khatib, 1987; Nakamura et al., 1987).

Here we discuss and describe trajectory-based control that we believe to strike a balance between the complexity and effectiveness in instances where desired motion trajectories are available or easy to obtain. Such control has been illustrated to be effective in the emerging applications, such as tracking of human motion from video (Vondrak et al., 2008).

2. Rigid body dynamics

Rigid bodies are solid structures that move in response to external forces exerted on them. They are characterized by mass density functions describing their volumes ("mass properties"), positions and orientations ("position information") in the world space and their time derivatives ("velocity information").

2.1 Body space, mass properties, position, orientation

Properties of rigid bodies are derived from an assumption that rigid bodies can be modeled as particle systems consisting of a large (infinite) number of particles constrained to remain at the same relative positions in the body spaces. Internal spatial interaction forces prevent bodies from changing their shapes and so as a result, any rigid body can only translate or rotate with respect to a fixed world frame of reference. This allows one to associate local coordinate frames with the bodies and define their shapes/volumes in terms of local *body spaces* that map to the world reference frame using rigid transformations.

We describe a volume of a rigid body by a *mass density function* $\rho: \mathbf{R}^3 \mapsto \mathbf{R}^+$ that determines the body's mass distribution over points \vec{r}^b in the body space. The density function is non-zero for points forming the body's shape and zero elsewhere and its moments characterize the body's response to the exerted forces. We are namely interested in *total mass* $m = \int \rho(\vec{r}^b) d\vec{r}^b$, *center of mass* $\vec{r}_{cm}^b = \int \frac{\vec{r}^b \rho(\vec{r}^b)}{m} d\vec{r}^b$, *principal moments of inertia* $I_{xx} = \int ((\vec{r}_y^b)^2 + (\vec{r}_z^b)^2) \rho(\vec{r}^b) d\vec{r}^b$, $I_{yy} = \int ((\vec{r}_x^b)^2 + (\vec{r}_z^b)^2) \rho(\vec{r}^b) d\vec{r}^b$, $I_{zz} = \int ((\vec{r}_x^b)^2 + (\vec{r}_y^b)^2) \rho(\vec{r}^b) d\vec{r}^b$ and *products of inertia* $I_{xy} = \int (\vec{r}_x^b \vec{r}_y^b) \rho(\vec{r}^b) d\vec{r}^b$, $I_{xz} = \int (\vec{r}_x^b \vec{r}_z^b) \rho(\vec{r}^b) d\vec{r}^b$, $I_{yz} = \int (\vec{r}_y^b \vec{r}_z^b) \rho(\vec{r}^b) d\vec{r}^b$ that we record into *inertia matrix*

$$I_{body} = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{pmatrix}.$$

To place a rigid body's volume in the world, we need to know the *mapping from the body space to the world space*. For that, we assume that the body's center of mass lies at the origin of the body space, $\vec{r}_{cm}^b = \vec{0}$, and construct a mapping $[R, \vec{x}]$ so that a point \vec{p}^b in the body space will get mapped to the world space point \vec{p} by applying a *rotation* R , represented by a 3×3 rotation matrix mapping body space axes to the world space axes (*orientation of the body in the world space*), followed by applying a *translation* \vec{x} that corresponds to the world space position of the body's center of mass (*position of the body in the world space*), $\vec{p} = R \cdot \vec{p}^b + \vec{x}$.

2.2 Velocity

Having placed the body in the world coordinate frame, we would like to characterize the motion of this body over time. To do so we need to compute time derivatives of the position and orientation of the body, i.e. $\frac{\partial}{\partial t}[R, \vec{x}]$. We decompose instantaneous motion over infinitesimally short time periods to the translational (*linear*) motion of the body's center of mass and a rotational (*angular*) motion of the body's volume. We first define *linear velocity* $\vec{v} = \dot{\vec{x}}$ as the time derivative of the rigid body's position \vec{x} , characterizing the instantaneous linear motion and describing the direction and speed of the body translation. Next, we describe the rotational motion as a rotation about a time varying axis that passes through the center of mass. We define *angular velocity* $\vec{\omega}$ as a world-space vector whose direction describes the instantaneous rotation axis and whose magnitude $[\text{rad} \cdot \text{s}^{-1}]$ defines the instantaneous rotation speed. Linear and angular velocities are related such that they can describe velocities of arbitrary points or vectors attached to the body. For example, if $\vec{r} = \vec{p} - \vec{x}$ is a vector between the point on the body, \vec{p} , the center of mass of the body, \vec{x} , then $\dot{\vec{r}} = \vec{\omega} \times \vec{r}$ and $\dot{\vec{p}} = \vec{v} + \vec{\omega} \times \vec{r}$. This can be used to derive a formula for \dot{R} that says $\dot{R} = \vec{\omega}^* \cdot R$, where $\vec{\omega}^*$ is a "cross-product matrix" such that $\vec{\omega}^* \cdot \vec{r} = \vec{\omega} \times \vec{r}$. It is worth noting that because \vec{p} is fixed in the body centric coordinate frame, so is the vector \vec{r} .

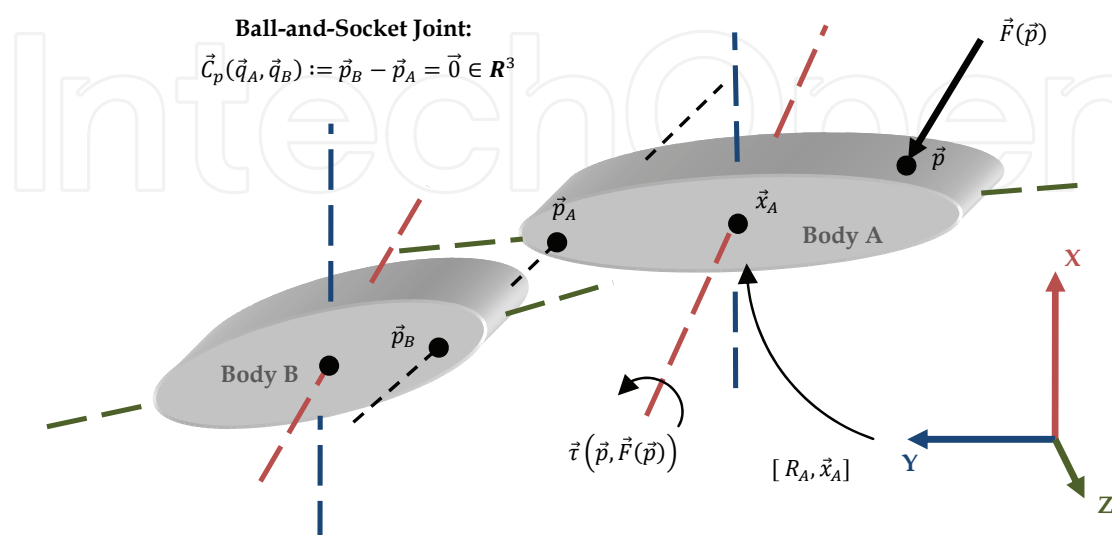


Fig. 2. Illustration of the two constrained bodies in motion.

2.3 Force

From previous section we have $\frac{\partial}{\partial t} [R, \vec{x}] = [\vec{\omega}^* \cdot R, \vec{v}]$ relating changes of the position and orientation to the values of the body's linear and angular velocities. Now, we would like to characterize how the linear and angular velocities of a rigid body change in response to forces exerted on the body. Intuitively, these changes should depend on the location where the force is applied as well as mass distribution over the body volume. So we need to know not only the directions and magnitudes of the exerted forces, but also the points at which these forces are applied.

To capture the effects for a single force $\vec{F}(\vec{p})$ acting at a world space point \vec{p} , we define a *force-torque pair* $[\vec{F}(\vec{p}), \vec{\tau}(\vec{p}, \vec{F}(\vec{p}))]$, where $\vec{\tau}(\vec{p}, \vec{F}(\vec{p})) = (\vec{p} - \vec{x}) \times \vec{F}(\vec{p})$ is the *torque* due to the force $\vec{F}(\vec{p})$. The torque can be imagined as a scale of the angular velocity $\vec{\omega}$ that the rigid body would gain if $\vec{F}(\vec{p})$ was the only force acting on the body and the force was exerted at \vec{p} . To capture the overall effects of all force-torque pairs $[\vec{F}_i, \vec{\tau}_i]$ due to all forces acting on the body, it is sufficient to maintain only the corresponding aggregate statistics: *total force* $\vec{F}_{total} = \sum_i \vec{F}_i$ and *total torque* $\vec{\tau}_{total} = \sum_i \vec{\tau}_i$ about the center of mass of the body, \vec{x} .

Now, we express the body's linear and angular velocities in the form of linear and angular momentums whose instantaneous changes can be directly related to the values of the total forces and torques acting on the body. The reason for doing so is that it is actually the momentums that remain unchanged when no forces act on the body, not the velocities. We define *linear momentum* $\vec{P} = m \cdot \vec{v}$ and *angular momentum* $\vec{L} = I \cdot \vec{\omega}$ where $I = R \cdot I_{body} \cdot R^T$. The relation between the velocity and force information is then given by derivatives of linear and angular momentum with respect to time, $\dot{\vec{P}} = \vec{F}_{total}$ and $\dot{\vec{L}} = \vec{\tau}_{total}$.

2.4 Equations of motion

We are now ready to present complete equations describing motion of a set of rigid bodies in Newtonian dynamics under the effect of forces. The equations are *first order ordinary differential equations* (ODEs). To simulate the system, one has to numerically integrate the equations of motion, which can be done by using standard numerical ODE solvers. We explore several formulations of the equations of motion below.

2.4.1 Momentum form

We start with the momentum form that makes the linear and angular momentum a part of a rigid body's state and builds directly upon the concepts presented in earlier sections. To make the body's state complete, only the position and orientation information has to be added to the state. Therefore, the *state* is described by a vector \vec{y} , $\vec{y} = (\vec{x}, R, \vec{P}, \vec{L})$, where \vec{x} is the position of the body's center of mass, R is the orientation of the body and \vec{P} and \vec{L} are the body's linear and angular momentums. The *equation of motion for the rigid body in the momentum form* is then given by $\frac{\partial \vec{y}}{\partial t} = (\vec{v}, \vec{\omega}^* \cdot R, \vec{F}_{total}, \vec{\tau}_{total})$, where \vec{F}_{total} and $\vec{\tau}_{total}$ are the total external force and torque exerted on the body and \vec{v} and $\vec{\omega}$ are auxiliary quantities derived from the state vector \vec{y} , $\vec{v} = m^{-1} \cdot \vec{P}$, $I = R \cdot I_{body} \cdot R^T$, $I^{-1} = R \cdot I_{body}^{-1} \cdot R^T$, $\vec{\omega} = I^{-1} \cdot \vec{L}$. If there are n rigid bodies in the system, the individual ODE equations are combined into a single ODE by concatenating the body states $\vec{y}_1, \dots, \vec{y}_n$ into a single state vector $\vec{y} = (\vec{y}_1, \dots, \vec{y}_n)$ and letting $\frac{\partial \vec{y}}{\partial t} = (\frac{\partial \vec{y}_1}{\partial t}, \dots, \frac{\partial \vec{y}_n}{\partial t})$.

2.4.2 Velocity form

As a conceptually more common alternative, the equations of motion can be reformulated so that linear and angular momentums in the state vector are replaced with linear and angular velocities. In this formulation, the state vector \vec{y} is defined as

$$\vec{y} = (\vec{x}, R, \vec{v}, \vec{\omega}) \quad (1)$$

To formulate the right-hand-side vector of the ODE, we need know time derivatives of the linear and angular velocities and relate them to external forces and torques. We define *linear acceleration* \vec{a} of a rigid body as the acceleration of the body's center of mass, that is, $\vec{a} = \dot{\vec{v}} = \ddot{\vec{x}}$, and because $\dot{\vec{P}} = \vec{F}_{total}$ we immediately get $\vec{a} = m^{-1} \cdot \vec{F}_{total}$. For the angular motion, we define *angular acceleration* $\vec{\alpha}$ as the time derivative of the body's angular velocity, $\vec{\alpha} = \dot{\vec{\omega}}$, and it can be shown that $\vec{\alpha} = I^{-1} \cdot (\vec{\tau}_{coriolis} + \vec{\tau}_{total})$, where $\vec{\tau}_{coriolis} = (I \times \vec{\omega}) \times \vec{\omega}$ is an implicit internal inertial (coriolis) torque due to body rotation and $\vec{\tau}_{total}$ is the total external torque applied on the body. This way we get the *equation of motion for a single¹ rigid body in the velocity form*

$$\frac{\partial \vec{y}}{\partial t} = \left(\vec{v}, \vec{\omega}^* \cdot R, m^{-1} \cdot \vec{F}_{total}, I^{-1} \cdot ((I \times \vec{\omega}) \times \vec{\omega} + \vec{\tau}_{total}) \right) \quad (2)$$

2.4.3 Generalized form

We now elaborate on the velocity-form of the equation of motion, define the notion of *generalized velocities and forces* and the concept of *mass matrices* for rigid bodies, which will allow us to treat rigid bodies as a kind of particles moving in \mathbf{R}^6 , simplifying many equations. We will call any block vector consisting of a block due to a linear quantity and a block due to the corresponding angular quantity a *generalized quantity*. That way, we obtain *generalized velocity* $\vec{v}_{gen} = (\vec{v}, \vec{\omega})$, *generalized acceleration* $\vec{a}_{gen} = (\vec{a}, \vec{\alpha})$, *generalized total external force* $\vec{F}_{gen}^{total} = (\vec{F}_{total}, \vec{\tau}_{total})$ and *generalized coriolis force* $\vec{F}_{gen}^{coriolis} = (\vec{0}, \vec{\tau}_{coriolis})$. In addition, we define *generalized position* $\vec{q} = (\vec{x}, \vec{R})$ that encodes both position of the body's center of mass and orientation in 3D space.

We now define the *mass matrix* M of a rigid body which is a 6×6 time-dependent matrix consisting of four 3×3 blocks encoding the body's mass properties,

$$M = \begin{pmatrix} m \cdot E & 0 \\ 0 & I \end{pmatrix}, \quad (3)$$

and E is a 3×3 identity matrix. From the previous section, we know that $m \cdot \vec{a} = \vec{F}_{total}$ and $I \cdot \vec{\alpha} = \vec{\tau}_{total} + \vec{\tau}_{coriolis}$ which can be rewritten using the mass matrix simply as $M \cdot \vec{a}_{gen} = \vec{F}_{gen}^{total} + \vec{F}_{gen}^{coriolis}$. Let's assume that the generalized coriolis force $\vec{F}_{gen}^{coriolis}$ is implicitly incorporated into the total generalized external force \vec{F}_{gen}^{total} and, to improve readability, let's remove the _{gen} subscripts and omit the "generalized" adjective whenever it is clear that the generalized notation is used. This lets us write

$$M \cdot \vec{a} = \vec{F}_{total} \quad (4)$$

¹ As for the momentum form, equation of motion for a set of n bodies is obtained by "cloning" the equation for a single body n -times.

which yields a relation between the total force \vec{F}_{total} and the total acceleration \vec{a} . Because the relation is linear, this equation also holds for *any* force \vec{F} acting on the body and the corresponding acceleration $\vec{a} = M^{-1} \cdot \vec{F}$ the body would gain in response to the application of \vec{F} ². The relation resembles Newton's Second Law for particles and rigid bodies can thus be imagined as special particles with time-varying masses M that move in \mathbf{R}^6 .

3. Constraints

One of the challenges one has to face in physical simulation is how to generate appropriate forces so that rigid bodies would move as desired. Instead of trying to generate these forces directly, we describe desired motion in terms of *motion constraints* on accelerations, velocities or positions of rigid bodies and then use *constraint solver* to solve for the forces. We still use the same equations of motion (and numerical solvers) to drive our bodies like before, but this time, we introduce *constraint forces* that implicitly act on constrained bodies so that given motion constraints are enforced. We study the approach of *Lagrange multiplier method* that handles each constraint in the same uniform way and allows to combine constraints automatically. Examples of constrained rigid bodies are given in Fig. 1.

In general, the motion constraint on the position or orientation of a body will subsequently result in the constraints on its velocity and acceleration (to ensure that there is no velocity or acceleration in the constrained direction, leading to violation of constraint after integration of the equations of motion); similarly a constraint on velocity will impose a constraint on the acceleration. We will discuss these implications in the following section. A first-order rigid body dynamics with impulsive formulation of forces (discussed in Section 3.3.1) allows one to ignore the acceleration constraints in favor of simplicity, but at expense of inability to support higher-order integration schemes.

3.1 Example: point-to-point equality constraint

Let's start with a motivational example. Imagine we are given two bodies and we want to enforce a position constraint that stipulates that point $\vec{p}_1 = \vec{x}_1 + \vec{r}_1$ attached to the first body is to coincide with a point $\vec{p}_2 = \vec{x}_2 + \vec{r}_2$ attached to the second body (see Fig. 2 where the two bodies are denoted as A and B), making the two bodies connected at $\vec{p}_1 = \vec{p}_2$ and preventing them from tearing apart. We can express this *position-level constraint* as a vector equation $\vec{C}_p(\vec{q}_1, \vec{q}_2) := \vec{p}_2 - \vec{p}_1 = \vec{0} \in \mathbf{R}^3$, defined in terms of generalized positions \vec{q}_1, \vec{q}_2 of the two bodies, such that all valid position pairs, for which the constraint is maintained, correspond to a manifold $\vec{C}_p(\vec{q}_1, \vec{q}_2) = \vec{0}$. Granted the constraint is maintained already, the goal is to compute an appropriate constraint force so that (\vec{q}_1, \vec{q}_2) stays on the manifold during the state update. Given the total external forces \vec{F}_1^{total} and \vec{F}_2^{total} acting on the two bodies, we will construct a constraint force such that it would cancel exactly those components of the \vec{F}_1^{total} and \vec{F}_2^{total} vectors that would make the bodies accelerate away from the manifold. To do this, we will reformulate our position-level constraint to a constraint on body accelerations and from that derive the constraint force. Our constraint formulation will give us a set of basis vectors that need be combined to get the constraint

² If \vec{F} refers to the total external force exerted on the body, coriolis force is assumed to be included in \vec{F} .

force. Appropriate coefficients of this combination are computed by solving a system of linear equations.

Let's assume that at the current time instant the bodies are positioned so that the constraint is maintained, that is, $\vec{C}_p = \vec{0}$. To make sure the constraint will also be maintained in the future, we have to enforce $\dot{\vec{C}}_p = \vec{0}$. Let's have a look at what $\dot{\vec{C}}_p$ looks like, $\dot{\vec{C}}_p = \frac{\partial}{\partial t}(\vec{p}_2 - \vec{p}_1) = \frac{\partial}{\partial t}(\vec{x}_2 + \vec{r}_2 - \vec{x}_1 - \vec{r}_1) = \dot{\vec{x}}_2 + \vec{\omega}_2 \times \vec{r}_2 - \dot{\vec{x}}_1 - \vec{\omega}_1 \times \vec{r}_1 = \dot{\vec{x}}_2 - \vec{r}_2 \times \vec{\omega}_2 - \dot{\vec{x}}_1 + \vec{r}_1 \times \vec{\omega}_1 = \dot{\vec{x}}_2 - \vec{r}_2^* \cdot \vec{\omega}_2 - \dot{\vec{x}}_1 + \vec{r}_1^* \cdot \vec{\omega}_1 = (-E \quad \vec{r}_1^*) \cdot \vec{v}_1 + (E \quad -\vec{r}_2^*) \cdot \vec{v}_2 = (-E \quad \vec{r}_1^* \quad E \quad -\vec{r}_2^*) \cdot \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \end{pmatrix} = (J_1 \quad J_2) \cdot \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \end{pmatrix}$, where J_1 and J_2 are 3×6 matrices called the Jacobian matrices due to the position constraint \vec{C}_p and the first and the second body. So we need to enforce another constraint $\vec{C}_v(\vec{v}_1, \vec{v}_2) := J_1 \cdot \vec{v}_1 + J_2 \cdot \vec{v}_2 = \vec{0}$, this time formulated in terms of generalized velocities \vec{v}_1, \vec{v}_2 . This is good because we were able to reformulate the original constraint specified in terms of generalized positions to a constraint specified in terms of generalized velocities.

Let's assume that the velocity constraint also holds, that is, $\vec{C}_v = \vec{0}$, and let's guarantee the velocity constraint will be maintained in the future by requesting $\dot{\vec{C}}_v = \vec{0}$ (this will also guarantee that the original position-level constraint will be maintained, because $\vec{C}_p = \vec{0}$ at the current time instant). We have $\dot{\vec{C}}_v = \frac{\partial}{\partial t}(J_1 \cdot \vec{v}_1 + J_2 \cdot \vec{v}_2) = J_1 \cdot \vec{a}_1 + J_2 \cdot \vec{a}_2 + \dot{J}_1 \cdot \vec{v}_1 + \dot{J}_2 \cdot \vec{v}_2$ and so we obtain a constraint $\vec{C}_a(\vec{a}_1, \vec{a}_2) := J_1 \cdot \vec{a}_1 + J_2 \cdot \vec{a}_2 - \vec{c} = \vec{0}$, where J_1 and J_2 are the Jacobian matrices defined above, \dot{J}_1 and \dot{J}_2 are their time derivatives and $\vec{c} = -\dot{J}_1 \cdot \vec{v}_1 - \dot{J}_2 \cdot \vec{v}_2$. This constraint is formulated directly in terms of generalized accelerations \vec{a}_1, \vec{a}_2 and because we already know the relation between accelerations and forces, this constrains the forces that can act on the two bodies. To complete the formulation of \vec{C}_a , we need to get the value of \vec{c} . It is usually easier to compute \vec{c} directly from $\dot{\vec{C}}_v$ rather than by computing the time derivatives of the Jacobian matrices. We can for example do, $\ddot{\vec{C}}_p = \dot{\vec{C}}_v = \frac{\partial}{\partial t}(-\dot{\vec{x}}_1 - \vec{\omega}_1 \times \vec{r}_1) + \frac{\partial}{\partial t}(\dot{\vec{x}}_2 - \vec{\omega}_2 \times \vec{r}_2) = (-\ddot{\vec{x}}_1 - \dot{\vec{\omega}}_1 \times \vec{r}_1 - \vec{\omega}_1 \times (\vec{\omega}_1 \times \vec{r}_1)) + (\ddot{\vec{x}}_2 + \dot{\vec{\omega}}_2 \times \vec{r}_2 + \vec{\omega}_2 \times (\vec{\omega}_2 \times \vec{r}_2)) = (-\ddot{\vec{x}}_1 + \vec{r}_1^* \cdot \dot{\vec{\omega}}_1 - \vec{\omega}_1 \times (\vec{\omega}_1 \times \vec{r}_1)) + (\ddot{\vec{x}}_2 - \vec{r}_2^* \cdot \dot{\vec{\omega}}_2 + \vec{\omega}_2 \times (\vec{\omega}_2 \times \vec{r}_2)) = (-E \quad \vec{r}_1^* \quad E \quad -\vec{r}_2^*) \cdot \begin{pmatrix} \vec{a}_1 \\ \vec{a}_2 \end{pmatrix} - \vec{\omega}_1 \times (\vec{\omega}_1 \times \vec{r}_1) + \vec{\omega}_2 \times (\vec{\omega}_2 \times \vec{r}_2)$ and obtain $\vec{c} = \vec{\omega}_1 \times (\vec{\omega}_1 \times \vec{r}_1) - \vec{\omega}_2 \times (\vec{\omega}_2 \times \vec{r}_2)$.

So given our original constraint $\vec{C}_p(\vec{q}_1, \vec{q}_2) := \vec{p}_2 - \vec{p}_1 = \vec{0}$ and assuming $\vec{C}_p = \vec{0}$ and $\dot{\vec{C}}_p = \vec{0}$ we were able to reduce the problem of maintaining $\vec{C}_p = \vec{0}$ to the problem of enforcing $\ddot{\vec{C}}_p = \vec{0}$ which is an *acceleration-level constraint* with $J_1 = (-E \quad \vec{r}_1^*)$, $J_2 = (E \quad -\vec{r}_2^*)$ and $\vec{c} = \vec{\omega}_1 \times (\vec{\omega}_1 \times \vec{r}_1) - \vec{\omega}_2 \times (\vec{\omega}_2 \times \vec{r}_2)$. We now need to compute the generalized constraint forces \vec{F}_1^c and \vec{F}_2^c to be applied to the first and second body, respectively. *Lagrange multiplier method* computes these forces as a linear combination of the rows of the Jacobian matrices (that are *known apriori*), $\vec{F}_1^c = J_1^T \cdot \vec{\lambda}$, $\vec{F}_2^c = J_2^T \cdot \vec{\lambda}$, and solves for the *unknown* coefficients (multipliers) $\vec{\lambda}$ in the combination so that $J_1 \cdot \vec{a}_1 + J_2 \cdot \vec{a}_2 = \vec{c}$ after the external forces \vec{F}_1^{total} and \vec{F}_2^{total} and constraint forces \vec{F}_1^c and \vec{F}_2^c were applied to the bodies. This can be imagined as follows. Each row of the three rows in $J_1 \cdot \vec{a}_1 + J_2 \cdot \vec{a}_2 = \vec{c} \in \mathbf{R}^3$ defines a hypersurface in

the space of points (\vec{a}_1, \vec{a}_2) and the (\vec{a}_1, \vec{a}_2) acceleration is valid if (\vec{a}_1, \vec{a}_2) lies on each of these hypersurfaces. Now, the normal of the j -th hypersurface equals the j -th row of $(J_1 \ J_2)$ and so in order to project (\vec{a}_1, \vec{a}_2) onto the j -th hypersurface, the force $\lambda_j \cdot (J_1)_j$ has to be applied to the first body and $\lambda_j \cdot (J_2)_j$ has to be applied to the second body.

Let's solve for the multipliers $\vec{\lambda}$. For that, let's concatenate individual vectors and matrices into global vectors and matrices characterizing the whole rigid body system, we get $\vec{a} = (\vec{a}_1, \vec{a}_2)$, $J = (J_1 \ J_2)$, $\vec{F}_{total} = (\vec{F}_1^{total}, \vec{F}_2^{total})$, $\vec{F}_c = J^T \cdot \vec{\lambda} = (\vec{F}_1^c, \vec{F}_2^c)$, $M = \begin{pmatrix} M_1 & 0 \\ 0 & M_2 \end{pmatrix}$ and $J \cdot \vec{a} = \vec{c}$. From the section on equations of motion, we get that the acceleration \vec{a} of the rigid body system after the total external force \vec{F}_{total} and constraint force \vec{F}_c are added to the system equals $\vec{a} = M^{-1} \cdot (\vec{F}_{total} + \vec{F}_c) = M^{-1} \cdot (\vec{F}_{total} + J^T \cdot \vec{\lambda}) = M^{-1} \cdot \vec{F}_{total} + M^{-1} \cdot J^T \cdot \vec{\lambda}$. This acceleration has to satisfy the constraint $J \cdot \vec{a} = \vec{c}$ and so $J \cdot M^{-1} \cdot \vec{F}_{total} + J \cdot M^{-1} \cdot J^T \cdot \vec{\lambda} = \vec{c}$, $(J \cdot M^{-1} \cdot J^T) \cdot \vec{\lambda} + (J \cdot M^{-1} \cdot \vec{F}_{total} - \vec{c}) = \vec{0}$, finally producing a system of linear equations $A \cdot \vec{\lambda} + \vec{b} = \vec{0}$, where $A = J \cdot M^{-1} \cdot J^T$ is a 3×3 matrix, $\vec{b} = J \cdot M^{-1} \cdot \vec{F}_{total} - \vec{c}$ is a 3×1 vector and $\vec{\lambda} \in \mathbf{R}^3$ are the multipliers to be solved for. Once $\vec{\lambda}$ are known, constraint force $\vec{F}_c = J^T \cdot \vec{\lambda} = (\vec{F}_1^c, \vec{F}_2^c)$ is applied to the bodies.

3.2 Acceleration constraints

We will now generalize the approach from the previous section for c constraints and n bodies. The index i will be used to index constraints, $i = 1, \dots, c$, and the index j will be used to index bodies, $j = 1, \dots, n$. Vectors $\vec{q} = (\vec{q}_1, \dots, \vec{q}_n)$, $\vec{v} = (\vec{v}_1, \dots, \vec{v}_n)$ and $\vec{a} = (\vec{a}_1, \dots, \vec{a}_n)$ will refer to the generalized position, velocity and acceleration of the rigid body system, $\vec{F}_{total} = (\vec{F}_1^{total}, \dots, \vec{F}_n^{total})$ will refer to the total external force exerted on the system and $\vec{F}_c = ((\vec{F}_c)_1, \dots, (\vec{F}_c)_n)$ will refer to the total constraint force exerted on the system due to all constraints.

Let M_j be the mass matrices of the individual bodies in the system. We then have $M_j \cdot \vec{a}_j = \vec{F}_j^{total}$ and so if M is a square block diagonal matrix with the individual matrices M_j on the diagonal, which we call the *mass matrix of the rigid body system*, we can relate the system acceleration \vec{a} due to the application of \vec{F}_{total} by $M \cdot \vec{a} = \vec{F}_{total}$, where

$$M = \begin{pmatrix} M_1 & 0 & \dots & 0 \\ 0 & M_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & M_n \end{pmatrix}.$$

Constraint i acts on two bodies A_i and B_i , has a dimensionality m_i and removes m_i degrees of freedom (DOFs) from the system. For example, if the two bodies are connected by a 3D revolute joint - $m_i = 3$, because the joint constrains position of body A_i with respect to B_i such that the two are affixed at the joint location (see Fig. 2). Note that while the constraint removes only 3 degrees of freedom, it affects both linear and angular properties of the system. A hinge joint will remove additional 2 degrees of freedom, allowing only rotational motion about a single axis with respect to the joint, resulting in a constraint of dimension $m_i = 5$, etc.

The constraint is characterized by a $m_i \times 6n$ matrix J_i of rank m_i called the constraint's Jacobian matrix consisting of n $m_i \times 6$ blocks due to individual bodies and a constraint

equation right-hand-side vector \vec{c}_i of length m_i . J_i has only two non-zero blocks, one due to the first constrained body A_i and one due to the second constrained body B_i , referred to by J_{i,A_i} and J_{i,B_i} . According to the Lagrange multiplier approach, the constraint is enforced by applying a constraint force $\vec{F}_c^i = J_i^T \cdot \vec{\lambda}_i = ((\vec{F}_c^i)_1, \dots, (\vec{F}_c^i)_n)$ to the rigid body system, determined by the values of m_i multipliers $\vec{\lambda}_i$. Each row $k = 1, \dots, m_i$ of J_i removes one DOF from the system and contributes to the constraint force \vec{F}_c^i by exerting a force $(\vec{\lambda}_i)_k \cdot (J_i)_k$ on the system. Due to the way J_i is defined, $(\vec{F}_c^i)_{A_i} = J_{i,A_i}^T \cdot \vec{\lambda}_i$ and $(\vec{F}_c^i)_{B_i} = J_{i,B_i}^T \cdot \vec{\lambda}_i$ are the only non-zero blocks of \vec{F}_c^i and $(\vec{F}_c^i)_{A_i}$ is the constraint force applied to the first body and $(\vec{F}_c^i)_{B_i}$ is the constraint force applied to the second body.

Let's stack the individual $m_i \times 6n$ Jacobian matrices J_i by rows to a single $m \times 6n$ Jacobian matrix J , where $m = \sum_i m_i$ is the total number of DOFs removed from the system. J is then a block matrix with $c \times n$ blocks whose non-zero blocks are given by J_{i,A_i} and J_{i,B_i} . Then the total constraint force \vec{F}_c exerted on the system equals $\vec{F}_c = \sum_i \vec{F}_c^i = J^T \cdot \vec{\lambda}$, where $\vec{\lambda} = (\vec{\lambda}_1, \dots, \vec{\lambda}_c)$ is a $m \times 1$ vector of Lagrange multipliers due to all constraints. Because constraints should not be conflicting, J is assumed to have full rank.

Let $A = J \cdot M^{-1} \cdot J^T$, $\vec{c} = (\vec{c}_1, \dots, \vec{c}_c)$ and $\vec{b} = J \cdot M^{-1} \cdot \vec{F}_{total} - \vec{c}$. Matrix A is a $m \times m$ matrix and can be treated as if it consisted of $c \times c$ blocks due to individual constraint pairs such that the value of the (i_1, i_2) -th block of size $m_{i_1} \times m_{i_2}$ due to the i_1 -th constraint and the i_2 -th constraint is given by $A_{i_1, i_2} = \sum_j J_{i_1, j} \cdot M_j^{-1} \cdot (J_{i_2, j})^T$. Because the individual matrices M_j and M_j^{-1} are positive definite, M and M^{-1} are positive definite and so because J is assumed to have full rank, A is also positive definite. We will use A_i (with slight abuse of notation) to denote the i -th block row of A due to constraint i . Vector \vec{b} is a vector of length m consisting of c blocks due to the individual constraints. We use \vec{b}_i to refer to the i -th block of \vec{b} of length m_i due to constraint i .

We will now discuss specific types of constraints. Each constraint i will generate a constraint force of the same form $\vec{F}_c^i = J_i^T \cdot \vec{\lambda}_i$ but different constraint types will lead to different conditions on the legal values of the multipliers $\vec{\lambda}$, essentially constraining the directions the constraint force can act along (can it push, can it pull or can it do both?).

3.2.1 Equality constraints

We define *acceleration level equality constraint i* as follows. The constraint acts on two bodies A_i and B_i , has a dimensionality m_i and is specified by two $m_i \times 6$ matrices J_{i,A_i} and J_{i,B_i} and a right-hand-side vector \vec{c}_i of length m_i . The constraint requests that $J_{i,A_i} \cdot \vec{a}_{A_i} + J_{i,B_i} \cdot \vec{a}_{B_i} = \vec{c}_i$ for accelerations \vec{a}_{A_i} and \vec{a}_{B_i} .

The J_{i,A_i} and J_{i,B_i} matrices are called the Jacobian blocks due to the first and the second body and are supposed to have full rank. This terminology stems from the fact that if the acceleration-level constraint implements a position-level constraint $\vec{C}_p(\vec{q}_{A_i}, \vec{q}_{B_i}) = \vec{0}$ or a velocity-level constraint $\vec{C}_v(\vec{v}_{A_i}, \vec{v}_{B_i}) = \vec{0}$ then $J_{i,A_i} = \frac{\partial \vec{C}_p}{\partial \vec{q}_{A_i}}$ and $J_{i,B_i} = \frac{\partial \vec{C}_p}{\partial \vec{q}_{B_i}}$ or $J_{i,A_i} = \frac{\partial \vec{C}_v}{\partial \vec{v}_{A_i}}$ and $J_{i,B_i} = \frac{\partial \vec{C}_v}{\partial \vec{v}_{B_i}}$. The constraint is an equality constraint because it is described by a linear equality.

Let's derive conditions on $\vec{\lambda}$ due to the acceleration level equality constraint i . Using our rigid body system dynamics equation, we get that the system acceleration \vec{a} after the total external force \vec{F}_{total} and total constraint force $\vec{F}_c = J^T \cdot \vec{\lambda}$ are applied to the system equals $\vec{a} = M^{-1} \cdot (\vec{F}_{total} + J^T \cdot \vec{\lambda})$. The constraint equation requests that $J_i \cdot \vec{a} - \vec{c}_i = \vec{0}$ which means that $J_i \cdot \vec{a} - \vec{c}_i = (J \cdot \vec{a} - \vec{c})_i = (J \cdot M^{-1} \cdot \vec{F}_{total} + J \cdot M^{-1} \cdot J^T \cdot \vec{\lambda} - \vec{c})_i = ((J \cdot M^{-1} \cdot J^T) \cdot \vec{\lambda} + (J \cdot M^{-1} \cdot \vec{F}_{total} - \vec{c}))_i = (A \cdot \vec{\lambda} + \vec{b})_i = A_i \cdot \vec{\lambda} + \vec{b}_i = \vec{0}$. Hence we get that equality constraint i requires that

$$A_i \cdot \vec{\lambda} + \vec{b}_i = \vec{0} \quad (5)$$

which is an equality constraint on the values of $\vec{\lambda}$.

3.2.2 Inequality constraints

Let's think of enforcing a different kind of constraint such that the equality sign $=$ in the constraint's formulation is replaced with either a greater-than-or-equal sign \geq or a less-than-or-equal sign \leq . For example, if $C_p(\vec{q}_1, \vec{q}_2)$ measures a distance of a ball from the ground plane, we might want to enforce a one-dimensional position constraint $C_p(\vec{q}_1, \vec{q}_2) \geq 0$ requesting that the ball lies above the ground. Assuming that both $C_p(\vec{q}_1, \vec{q}_2) = 0$ and $\dot{C}_p(\vec{q}_1, \vec{q}_2) = 0$ (the ball rests on the ground), the constraint can be implemented by maintaining $\ddot{C}_p(\vec{q}_1, \vec{q}_2) \geq 0$, which is an acceleration-level greater-or-equal constraint.

3.2.2.1 Greater-or-equal constraints

We define *acceleration level greater-or-equal constraint i* as follows. The constraint acts on two bodies A_i and B_i , has a dimensionality m_i and is specified by two $m_i \times 6$ matrices J_{i,A_i} and J_{i,B_i} and a right-hand-side vector \vec{c}_i of length m_i . The constraint requests that $J_{i,A_i} \cdot \vec{a}_{A_i} + J_{i,B_i} \cdot \vec{a}_{B_i} \geq \vec{c}_i$ for accelerations \vec{a}_{A_i} and \vec{a}_{B_i} .

Let's present conditions on $\vec{\lambda}$ due to the acceleration level greater-or-equal constraint i . Similarly to the equality case, $J_{i,A_i} \cdot \vec{a}_{A_i} + J_{i,B_i} \cdot \vec{a}_{B_i} \geq \vec{c}_i$ can be rewritten as (1) $J_{i,A_i} \cdot \vec{a}_{A_i} + J_{i,B_i} \cdot \vec{a}_{B_i} - \vec{c}_i = J_i \cdot \vec{a} - \vec{c}_i = A_i \cdot \vec{\lambda} + \vec{b}_i \geq \vec{0}$, which is an inequality greater-or-equal constraint on the values of $\vec{\lambda}$. Now, let's recall that in *Lagrange multiplier approach*, the goal of \vec{F}_c^i is to cancel those components of \vec{F}_{total} that would make the bodies accelerate towards invalid states. In the case of an equality constraint, the bodies were restricted to remain on the intersections of the hypersurfaces due to the constraint's DOFs and \vec{F}_c^i cancelled accelerations along the directions of the hypersurface normals. In the case of a greater-or-equal constraint, however, the bodies can move away from a hypersurface along the direction of the hypersurface's normal, but not in the opposite direction. In other words, positive accelerations along the positive directions of the normals are unconstrained and therefore (2) $\vec{\lambda}_i \geq \vec{0}$ (the constraint force can not pull the bodies back to the hypersurface). In addition, (3) if the bodies are already accelerating to the front of the hypersurface k , $(J_{i,A_i} \cdot \vec{a}_{A_i} + J_{i,B_i} \cdot \vec{a}_{B_i} - \vec{c}_i)_k > 0$, then the constraint force due to that hypersurface must vanish, that is $(\vec{\lambda}_i)_k = 0$, so that no energy would be added to the system (constraint force is as "lazy" as possible). These conditions can be restated in terms of the i -th block row of matrix A and the i -th block of vector \vec{b} as follows,

$$\begin{aligned}
A_i \cdot \vec{\lambda} + \vec{b}_i &\geq \vec{0} \\
\vec{\lambda}_i &\geq \vec{0} \\
(A_i \cdot \vec{\lambda} + \vec{b}_i) \cdot \vec{\lambda}_i &= 0,
\end{aligned} \tag{6}$$

where $(A_i \cdot \vec{\lambda} + \vec{b}_i) \cdot \vec{\lambda}_i = \sum_{k=1}^{m_i} (A_i \cdot \vec{\lambda} + \vec{b}_i)_k \cdot (\vec{\lambda}_i)_k = 0$ in fact means that $(A_i \cdot \vec{\lambda} + \vec{b}_i)_k \cdot (\vec{\lambda}_i)_k$ for $1 \leq k \leq m_i$ because both the products have to be positive. It is said that the components of $A_i \cdot \vec{\lambda} + \vec{b}_i$ are *complementary* to the corresponding components of $\vec{\lambda}_i$.

3.2.2.2 Less-or-equal constraints

We define *acceleration level less-or-equal constraint i* as follows. The constraint acts on two bodies A_i and B_i , has a dimensionality m_i and is specified by two $m_i \times 6$ matrices J_{i,A_i} and J_{i,B_i} and a right-hand-side vector \vec{c}_i of length m_i . The constraint requests that $J_{i,A_i} \cdot \vec{a}_{A_i} + J_{i,B_i} \cdot \vec{a}_{B_i} \leq \vec{c}_i$ for accelerations \vec{a}_{A_i} and \vec{a}_{B_i} .

Analogously to the previous case, we obtain the following set of conditions on multipliers $\vec{\lambda}$ due to the acceleration level less-or-equal constraint i . In addition to the condition $J_i \cdot \vec{a} - \vec{c}_i = A_i \cdot \vec{\lambda} + \vec{b}_i \leq \vec{0}$, multipliers due to constraint i have to be negative and complementary to $\vec{\lambda}_i$,

$$\begin{aligned}
A_i \cdot \vec{\lambda} + \vec{b}_i &\leq \vec{0} \\
\vec{\lambda}_i &\leq \vec{0} \\
(A_i \cdot \vec{\lambda} + \vec{b}_i) \cdot \vec{\lambda}_i &= 0.
\end{aligned} \tag{7}$$

Less-or-equal constraints i can trivially be converted to greater-or-equal constraints by negating the Jacobian blocks and the right-hand-side vector \vec{c}_i and so they do not have to be handled as a special case.

3.2.3 Bounded equality constraints

Let's suppose we want to implement a one-dimensional constraint that would behave like an equality constraint $J_i \cdot \vec{a} = \vec{c}_i$ such that the constraint would break if the magnitude $\|J_i^T\| \cdot |(\vec{\lambda}_i)_1|$ of the constraint force $\vec{F}_c^i = J_i^T \cdot \vec{\lambda}_i$ required to maintain the constraint exceeds a certain limit. Such a capability could, for example, be used for the implementation of various kinds of motors with limited power. Now, because $\|J_i^T\|$ is known, limiting the force magnitude (in this case) is equivalent to specifying the lower and upper bound on the value of the multiplier $(\vec{\lambda}_i)_1$. Hence, without loss of generality we can assume the bounds on $\vec{\lambda}_i$ are given instead. In the general case of a multi-dimensional constraint, we assume that each multiplier has its own bounds, independent of the values of other multipliers, so that the problem of solving for $\vec{\lambda}$ remains tractable.

We define *acceleration level bounded equality constraint i* as follows. The constraint acts on two bodies A_i and B_i , has a dimensionality m_i and is specified by two $m_i \times 6$ matrices J_{i,A_i} and J_{i,B_i} , a right-hand-side vector \vec{c}_i of length m_i and $\vec{\lambda}_i$ bounds $\vec{\lambda}_i^{lo} \leq \vec{0}$ and $\vec{\lambda}_i^{hi} \geq \vec{0}$. The constraint requests that $(\vec{\lambda}_i^{lo})_k \leq (\vec{\lambda}_i)_k \leq (\vec{\lambda}_i^{hi})_k$ and implements the equality constraint $J_{i,A_i} \cdot \vec{a}_{A_i} + J_{i,B_i} \cdot \vec{a}_{B_i} = \vec{c}_i$ for accelerations \vec{a}_{A_i} and \vec{a}_{B_i} subject to constraint force limits given by $\vec{\lambda}_i^{lo}$ and $\vec{\lambda}_i^{hi}$.

We will now elaborate on what constraint force limits due to the acceleration level bounded equality constraint i really mean and what the corresponding conditions on $\vec{\lambda}$ look like. Following up on the hypersurface interpretation of the equality constraint $J_i \cdot \vec{a} - \vec{c}_i = \vec{0}$, if the bodies are to move off the hypersurface k due to the k -th constraint DOF in the direction of the surface normal, a negative $(\vec{\lambda}_i)_k$ is required to cancel the acceleration. Now, if the value of $(\vec{\lambda}_i)_k$ required to fully cancel the acceleration is less than the allowed lower limit $(\vec{\lambda}_i^{lo})_k$, clamped $(\vec{\lambda}_i)_k \geq (\vec{\lambda}_i^{lo})_k$ would not yield a constraint force strong enough to cancel the prohibited acceleration and in the end $J_i \cdot \vec{a} - \vec{c}_i > \vec{0}$. Similarly, if the bodies are to move off the hypersurface in the opposite direction, a positive $(\vec{\lambda}_i)_k$ is required to cancel the acceleration. If $(\vec{\lambda}_i)_k$ is clamped such that $(\vec{\lambda}_i)_k \leq (\vec{\lambda}_i^{hi})_k$ and the acceleration is not cancelled fully then $J_i \cdot \vec{a} - \vec{c}_i < \vec{0}$. Putting this discussion into equations and assuming $(\vec{\lambda}_i^{lo})_k \leq 0$ and $(\vec{\lambda}_i^{hi})_k \geq 0$, we get

$$\begin{aligned}
 (\vec{\lambda}_i^{lo})_k &\leq (\vec{\lambda}_i)_k \leq (\vec{\lambda}_i^{hi})_k \\
 (\vec{\lambda}_i)_k = (\vec{\lambda}_i^{lo})_k &\Rightarrow (A_i \cdot \vec{\lambda} + \vec{b}_i)_k \geq 0 \\
 (\vec{\lambda}_i)_k = (\vec{\lambda}_i^{hi})_k &\Rightarrow (A_i \cdot \vec{\lambda} + \vec{b}_i)_k \leq 0 \\
 (\vec{\lambda}_i^{lo})_k < (\vec{\lambda}_i)_k < (\vec{\lambda}_i^{hi})_k &\Rightarrow (A_i \cdot \vec{\lambda} + \vec{b}_i)_k = 0.
 \end{aligned} \tag{8}$$

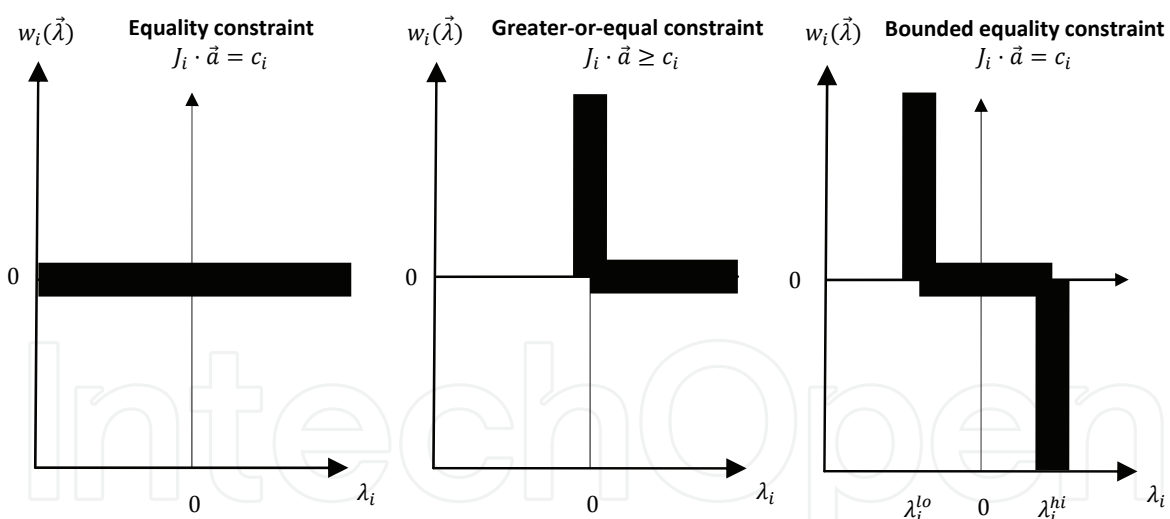


Fig. 3. Visualization of complementarity conditions on the pairs $(\lambda_i, w_i(\vec{\lambda}))$ due to different kinds of one dimensional constraints i , where $w_i(\vec{\lambda}) := A_i \cdot \vec{\lambda} + b_i = J_i \cdot \vec{a} - c_i$. Thick lines indicate permissible values for the $(\lambda_i, w_i(\vec{\lambda}))$ pairs. As can be seen, equality constraint requests $w_i(\vec{\lambda})$ to be zero and lets λ_i take an arbitrary value. Greater-or-equal constraint requests both $w_i(\vec{\lambda})$ and λ_i to be non-negative and complementary to each other. Bounded equality constraint generalizes the two previous cases by introducing explicit limits $\lambda_i^{lo} \leq 0$ and $\lambda_i^{hi} \geq 0$ on the values of λ_i . For improved readability, $\vec{\cdot}$ accents have been removed from one-dimensional vectors related to the constraint i .

Bounded equality constraints are generalization of both inequality and equality constraints. For example, if we set $\vec{\lambda}_i^{lo} = \vec{0}$ and $\vec{\lambda}_i^{hi} = \vec{\infty}$ then the bounded equality constraint i turns to a greater-or-equal constraint i with the same Jacobian blocks and right-hand-side vector \vec{c}_i . Similarly, by setting $\vec{\lambda}_i^{lo} = -\vec{\infty}$ and $\vec{\lambda}_i^{hi} = \vec{0}$, the constraint turns to a less-or-equal constraint. Finally, by setting $\vec{\lambda}_i^{lo} = -\vec{\infty}$ and $\vec{\lambda}_i^{hi} = \vec{\infty}$, the constraint turns to an unbounded equality constraint.

3.2.4 Reduction to LCP

In the previous section we have discussed several constraint types and showed what conditions on the multipliers $\vec{\lambda}$ they impose. Our goal is now to solve for $\vec{\lambda}$ obeying the presented conditions so that the constraint force $\vec{F}_c = J^T \cdot \vec{\lambda}$ could be exerted on the system. As it turns out, the problem of solving for $\vec{\lambda}$ is equivalent to solving of specific kinds of *linear complementarity problems (LCPs)* for which efficient algorithms exist and so we can compute $\vec{\lambda}$ by using a LCP solver, (Smith, 2004; Vondrak, 2006; Cline, 2002). To simplify the discussion, we assume that every inequality and bounded equality constraint i is one-dimensional, $m_i = 1$. As a result, we can simply write λ_i instead of $(\vec{\lambda}_i)_1$, etc.

If all the constraints are unbounded equalities, the corresponding conditions on $\vec{\lambda}$ are given by $A \cdot \vec{\lambda} + \vec{b} = \vec{0}$ which is a linear system that can be solved efficiently by standard factorization techniques. If all constraints are greater-or-equal constraints, we get a *pure linear complementarity problem* of the form $A \cdot \vec{\lambda} + \vec{b} \geq \vec{0}$, $\vec{\lambda} \geq \vec{0}$, $\vec{\lambda} \cdot (A \cdot \vec{\lambda} + \vec{b}) = \vec{0}$, which can be solved by a standard LCP solver. If there are k unbounded equality constraints and $c - k$ greater-or-equal constraints, we get a *mixed linear complementarity problem* $A_{eq} \cdot \vec{\lambda} + \vec{b}_{eq} = \vec{0}$, $A_{ineq} \cdot \vec{\lambda} + \vec{b}_{ineq} \geq \vec{0}$, $\vec{\lambda}_{ineq} \geq \vec{0}$, $\vec{\lambda}_{ineq} \cdot (A_{ineq} \cdot \vec{\lambda} + \vec{b}_{ineq}) = 0$, where A_{eq}, \vec{b}_{eq} denotes the rows of A, \vec{b} due to equality constraints and A_{ineq}, \vec{b}_{ineq} denotes the rows of A, \vec{b} due to inequality constraints. Mixed LCPs can be solved by mixed LCP solvers. Finally, if there are k unbounded equality constraints and $c - k$ bounded equality-constraints (including inequality constraints i with appropriately set $\vec{\lambda}_i$ limits), we get a *lo-hi linear complementarity problem* $A_{eq} \cdot \vec{\lambda} + \vec{b}_{eq} = \vec{0}, \lambda_i^{lo} \leq \lambda_i \leq \lambda_i^{hi}, \lambda_i = \lambda_i^{lo} \Rightarrow A_i \cdot \vec{\lambda} + b_i \geq 0, \lambda_i = \lambda_i^{hi} \Rightarrow A_i \cdot \vec{\lambda} + b_i \leq 0, \lambda_i^{lo} < \lambda_i < \lambda_i^{hi} \Rightarrow A_i \cdot \vec{\lambda} + b_i = 0$, where i indexes unbounded equality and inequality constraints. This is the most general form that can handle all constraint forms we have discussed and can also be solved efficiently.

3.3 Velocity constraints

So far we have discussed how constraints can be implemented on the accelerations. It is useful, however, to specify constraints on the velocities as well. Let's recall the example with the ball and the ground plane where the goal is to enforce a one-dimensional position-level constraint $C_p(\vec{q}_1, \vec{q}_2) \geq 0$ stipulating that the ball has to stay above the ground. Now, if $C_p(\vec{q}_1(t), \vec{q}_2(t)) = 0$ and $\dot{C}_p(\vec{q}_1(t), \vec{q}_2(t)) < 0$ at the current time t (the ball strikes the ground plane) then $C_p(\vec{q}_1(t + \epsilon), \vec{q}_2(t + \epsilon)) < 0$ at the time instant $t + \epsilon$ regardless of accelerations at time t for a sufficiently small ϵ . In order to ensure that the constraint is maintained at $t + \epsilon$, velocities at time t have to change so that $\dot{C}_p(\vec{q}_1(t), \vec{q}_2(t)) \geq 0$. This, however, is a constraint on the velocity.

3.3.1 Impulsive dynamics

We will now outline the concept of impulsive forces and first-order rigid body dynamics. With regular forces, the effects of forces on positions and orientations of rigid bodies are determined by second-order (Newtonian) dynamics in which velocities change through the integration of forces while positions change through the integration of velocities. With impulsive forces, the effects of forces on positions and orientations are determined by first-order (impulsive) dynamics in which velocities change directly through the application of impulsive forces and positions change through the integration of velocities.

We postulate *impulsive force* \vec{J}_F as a force with “units of momentum”. If \vec{P} and \vec{L} are the linear and angular momentums of a rigid body and \vec{J}_F is applied to the body at the world space position \vec{r} , then the linear momentum \vec{P} changes by the value $\Delta\vec{P} = \vec{J}_F$ and the angular momentum \vec{L} changes by the value $\Delta\vec{L} = \vec{J}_\tau$, where $\vec{J}_\tau = (\vec{r} - \vec{x}) \times \vec{J}_F$ is *impulsive torque* due to the impulsive force \vec{J}_F . Impulsive forces and torques can be seen as “ordinary” forces and torques that directly change the body’s linear and angular momentums, instead of affecting their time derivatives.

Similarly to the second-order dynamics, we couple linear and corresponding angular quantities to generalized quantities. That way, we obtain *generalized momentum* $\vec{F}_{imp}^{total} = (\vec{P}, \vec{L})$ and *generalized impulsive force (impulse)* $\vec{F}_{imp} = (\vec{J}_F, \vec{J}_\tau)$. Then if M is the mass matrix of the rigid body and \vec{v} is the body’s generalized velocity, we immediately get $M \cdot \vec{v} = \vec{F}_{imp}^{total}$ from the definition of the linear and angular momentum. Moreover, our momentum update rules state that the change $\Delta\vec{v}$ of generalized velocity \vec{v} due to the application of the generalized impulse \vec{F}_{imp} equals $\Delta\vec{v} = M^{-1} \cdot \vec{F}_{imp}$. Therefore the first-order dynamics relating velocities \vec{v} to impulses \vec{F}_{imp} is given by

$$M \cdot \vec{v} = \vec{F}_{imp} \quad (9)$$

and \vec{F}_{imp}^{total} can be seen as a generalized *total external impulse* acting on the body that consists of the only term – the inertial term (\vec{P}, \vec{L}) . This directly compares to the case of second-order dynamics that relates accelerations \vec{a} to forces \vec{F} by $M \cdot \vec{a} = \vec{F}$.

If we have a set of n rigid bodies with mass matrices M_1, \dots, M_n , generalized velocities $\vec{v}_1, \dots, \vec{v}_n$ and total external impulses $(\vec{F}_{imp}^{total})_1, \dots, (\vec{F}_{imp}^{total})_n$ then the first-order dynamics of the system is given by $M \cdot \vec{v} = \vec{F}_{imp}^{total}$, where M is a mass matrix of the system made of M_1, \dots, M_n , $\vec{v} = (\vec{v}_1, \dots, \vec{v}_n)$ and $\vec{F}_{imp}^{total} = ((\vec{F}_{imp}^{total})_1, \dots, (\vec{F}_{imp}^{total})_n)$. Analogously to the acceleration case, we call \vec{v} the velocity of the system and \vec{F}_{imp}^{total} the total external impulse exerted on the system (system momentum).

3.3.2 Constraints

We can now transfer everything we know about acceleration-level constraints, defined with respect to accelerations and forces, to the realm of velocity-level constraints, defined with respect to velocities and impulsive forces. There is no need to do any derivations because acceleration-level formulation of rigid body dynamics exactly corresponds to the velocity-level formulation of the impulsive dynamics. The only differences are due to the fact that we will now work with system velocities \vec{v} , impulsive constraint forces \vec{F}_{imp}^c and

momentums \vec{F}_{imp}^{total} instead of accelerations \vec{a} , constraint forces \vec{F}_c and total external forces \vec{F}_{total} . In consequence, the same algorithms can be used to implement velocity constraints. We define *velocity level constraint* i as follows. The constraint acts on two bodies A_i and B_i , has a dimensionality m_i and is specified by two $m_i \times 6$ matrices J_{i,A_i} and J_{i,B_i} and a right-hand-side vector \vec{k}_i of length m_i . The constraint requests either $J_{i,A_i} \cdot \vec{v}_{A_i} + J_{i,B_i} \cdot \vec{v}_{B_i} = \vec{k}_i$, $J_{i,A_i} \cdot \vec{v}_{A_i} + J_{i,B_i} \cdot \vec{v}_{B_i} \leq \vec{k}_i$ or $J_{i,A_i} \cdot \vec{v}_{A_i} + J_{i,B_i} \cdot \vec{v}_{B_i} \geq \vec{k}_i$ and is implemented by exerting a constraint impulse $(\vec{F}_c^i)_{imp} = J_i^T \cdot \vec{\lambda}_i$ determined by the values of multipliers $\vec{\lambda}_i$. In addition, if bounds on the valid multiplier values $\vec{\lambda}_i^{lo} \leq \vec{0}$ and $\vec{\lambda}_i^{hi} \geq \vec{0}$ are provided, then the constraint describes a *bounded equality constraint* i that requests $(\vec{\lambda}_i^{lo})_k \leq (\vec{\lambda}_i)_k \leq (\vec{\lambda}_i^{hi})_k$ and implements the equality constraint $J_{i,A_i} \cdot \vec{v}_{A_i} + J_{i,B_i} \cdot \vec{v}_{B_i} = \vec{k}_i$ for velocities \vec{v}_{A_i} and \vec{v}_{B_i} subject to constraint impulse limits given by $\vec{\lambda}_i^{lo}$ and $\vec{\lambda}_i^{hi}$. Multipliers $\vec{\lambda}$ can be computed by solving the same LCP problems like before. If there are c constraints, we will get $A = J \cdot M^{-1} \cdot J^T$ and $\vec{b} = J \cdot M^{-1} \cdot \vec{F}_{imp}^{total} - \vec{k}$, where $\vec{k} = (\vec{k}_1, \dots, \vec{k}_c)$.

3.4 Position constraints

Motion control constraints are most often specified on the position level because it is the natural way of expressing desired motion. In the earlier section, we have already discussed how position level constraints can be implemented either on the acceleration or velocity level, but this time, we will do it more thoroughly and will also show how prior constraint errors due to numerical inaccuracies could be reduced during simulation.

We never enforce constraints directly on the position level. Position level enforcement would require use of custom equations of motion specific to the set of constraints. As a result equations would have to change each time the constraint set is updated. For the rest of the section, we will assume we have n rigid bodies and c position-level constraints.

We define *position level constraint* i as follows. The constraint acts on two bodies A_i and B_i , has a dimensionality m_i and is specified by a function $\vec{C}_p^i(\vec{q}_{A_i}, \vec{q}_{B_i}) \in \mathbf{R}^{m_i}$ that is differentiable with respect to time so that its velocity level and acceleration level formulations (consistent with our prior definitions) can be obtained by differentiation. *Position level equality constraint* i requests that $\vec{C}_p^i(\vec{q}_{A_i}, \vec{q}_{B_i}) = \vec{0}$ for generalized positions \vec{q}_{A_i} and \vec{q}_{B_i} and the value of $\vec{C}_p^i(\vec{q}_{A_i}, \vec{q}_{B_i})$ can intuitively be thought of as a measurement of the position error for bodies at the position configuration $(\vec{q}_{A_i}, \vec{q}_{B_i})$. *Position level greater-or-equal constraint* i requests that $\vec{C}_p^i(\vec{q}_{A_i}, \vec{q}_{B_i}) \geq \vec{0}$ and *position level less-or-equal constraint* i requests that $\vec{C}_p^i(\vec{q}_{A_i}, \vec{q}_{B_i}) \leq \vec{0}$.

3.4.1 Acceleration or velocity level

We use constraint forces to implement position level constraints in an incremental way. We start from an initial state that is consistent with the constraint formulation (such that positions and velocities are valid with respect to the position level and velocity level formulations of the constraints) and then apply constraint forces to ensure that the velocity level and position level constraints remain maintained. Alternatively, we start from a state that is consistent with the position level formulations and then apply constraint impulses to ensure that the position level constraints remain maintained.

Please note that whenever an impulse is applied to a body, its velocity changes. In consequence, conditions that have to be met so that a particular constraint could be

implemented on the acceleration level need no longer be valid after the impulse is applied and so it cannot be reliably determined in advance which constraints can be implemented on the acceleration level. To address this issue, we implement all constraints on the velocity level *whenever there is at least one position constraint that has to be implemented on the velocity level*.

3.4.2 Equality constraints with stabilization

Consider the position level equality constraint $\vec{C}_p^i(\vec{q}_{A_i}, \vec{q}_{B_i}) = \vec{0}$. By differentiating $\vec{C}_p^i(\vec{q}_{A_i}, \vec{q}_{B_i}) = \vec{0}$ with respect to time, we get a corresponding velocity level formulation of the position constraint in the form of $\vec{C}_v^i(\vec{v}_{A_i}, \vec{v}_{B_i}) = \vec{0}$, where $\vec{C}_v^i(\vec{v}_{A_i}, \vec{v}_{B_i}) = \frac{\partial}{\partial t} \vec{C}_p^i(\vec{q}_{A_i}, \vec{q}_{B_i}) = J_{i,A_i} \cdot \vec{v}_{A_i} + J_{i,B_i} \cdot \vec{v}_{B_i}$. By differentiating this velocity constraint, we get a corresponding acceleration level formulation $\vec{C}_a^i(\vec{a}_{A_i}, \vec{a}_{B_i}) = \vec{0}$, where $\vec{C}_a^i(\vec{a}_{A_i}, \vec{a}_{B_i}) = \frac{\partial}{\partial t} \vec{C}_v^i(\vec{v}_{A_i}, \vec{v}_{B_i}) = J_{i,A_i} \cdot \vec{a}_{A_i} + J_{i,B_i} \cdot \vec{a}_{B_i} - \vec{c}_i$ and $\vec{c}_i = -\dot{J}_{i,A_i} \cdot \vec{v}_{A_i} - \dot{J}_{i,B_i} \cdot \vec{v}_{B_i}$. The position level constraint i $\vec{C}_p^i = \vec{0}$ can thus be implemented *incrementally* either (1) on the acceleration level, by starting from a state where $\vec{C}_p^i = \dot{\vec{C}}_p^i = \vec{0}$ and applying constraint forces so that $\ddot{\vec{C}}_p^i = \vec{0}$ or (2) on the velocity level, by starting from a state where $\vec{C}_p^i = \vec{0}$ and applying constraint impulses so that $\dot{\vec{C}}_p^i = \vec{0}$. In the first case, constraint forces are applied under the assumption that $\vec{C}_p^i = \dot{\vec{C}}_p^i = \vec{0}$, while in the second case, constraint impulses are applied under the assumption that $\dot{\vec{C}}_p^i = \vec{0}$. In practice, however, these assumptions often do not hold for various pragmatic reasons. For example, the numerical solver that integrates the equations of motion incurs an integration error or constraint forces are computed with an insufficient precision.

Let's assume we implement the position level constraint i on the velocity level. If the constraint is currently broken, that is $\vec{C}_p^i \neq \vec{0}$, we want to generate a constraint impulse so that the constraint error \vec{C}_p^i will be driven towards a zero vector. This is called *constraint stabilization*. Fortunately, simple stabilization can be implemented by following a procedure suggested in (Cline, 2002). Instead of requiring that $\dot{\vec{C}}_p^i = 0$, we can require that

$$\dot{\vec{C}}_p^i = -\vec{C}_p^i \cdot \alpha, \quad (10)$$

where α is a small positive value (dependent on the integration step size) that determines the speed with which the constraint is stabilized. Then, if t is the current time, we have $\vec{C}_p^i(t + \Delta t) \approx \vec{C}_p^i(t) + \Delta t \cdot \dot{\vec{C}}_p^i(t) = \vec{C}_p^i(t) \cdot (1 - \Delta t \cdot \alpha)$ and so we can reduce the position error by simply biasing the request on the desired velocity.

Analogously to the previous case, if we implement the position level constraint i on the acceleration level, we need to reduce both the position error \vec{C}_p^i as well as velocity error $\dot{\vec{C}}_p^i$. That could be done by biasing the request on the desired acceleration $\ddot{\vec{C}}_p^i$. Instead of requiring that $\ddot{\vec{C}}_p^i = \vec{0}$ we can require

$$\ddot{\vec{C}}_p^i = -\vec{C}_p^i \cdot \alpha - \dot{\vec{C}}_p^i \cdot \beta, \quad (11)$$

where α and β are positive constants. Because $\dot{\vec{C}}_p^i = J_i \cdot \vec{v}$ we get $\ddot{\vec{C}}_p^i = -\vec{C}_p^i \cdot \alpha - J_i \cdot \vec{v} \cdot \beta$.

Plugging these equations into our constraint definitions, we can therefore implement the position level equality constraint i with stabilization by submitting either the velocity level

equality constraint $J_{i,A_i} \cdot \vec{v}_{A_i} + J_{i,B_i} \cdot \vec{v}_{B_i} = -\vec{C}_p^i \cdot \alpha$ or the acceleration level equality constraint $J_{i,A_i} \cdot \vec{a}_{A_i} + J_{i,B_i} \cdot \vec{a}_{B_i} = -\vec{C}_p^i \cdot \alpha - J_i \cdot \vec{v} \cdot \beta - \dot{J}_i \cdot \vec{v}$. Moreover, if we want to implement powering limits, these constraints can be submitted as bounded equality constraints with appropriate force limits.

3.4.3 Inequality constraints with stabilization

Let's assume for simplicity that we work with a one-dimensional position level greater-or-equal constraint $C_p^i \geq 0$, like the one that would stipulate that the ball lies above the ground. This constraint is fundamentally different from a equality constraint because no velocity or acceleration constraints are actually imposed until the constrained bodies reach the boundary $\{(\vec{q}_{A_i}, \vec{q}_{B_i}) \mid C_p^i(\vec{q}_{A_i}, \vec{q}_{B_i}) = 0\}$ of the set of valid positions $\{(\vec{q}_{A_i}, \vec{q}_{B_i}) \mid C_p^i(\vec{q}_{A_i}, \vec{q}_{B_i}) \geq 0\}$ in the position space. We will now discuss three cases, depending on whether $C_p^i > 0$, $C_p^i = 0$ or $C_p^i < 0$ at the current state.

If $C_p^i > 0$ then the bodies did not reach the boundary of the set of valid positions and so no constraints will be imposed. This corresponds to the case when the ball is in the air and does not touch the ground.

If $C_p^i = 0$ then the bodies lie at the boundary of the set of valid positions (the ball touches the ground) and $\dot{C}_p^i \geq 0$ has to be enforced in order to maintain $C_p^i \geq 0$. This $\dot{C}_p^i \geq 0$ constraint can be implemented (1) on the acceleration level by submitting $\ddot{C}_p^i \geq 0$ if $\dot{C}_p^i = 0$, (2) on the acceleration level by ignoring the constraint if $\dot{C}_p^i > 0$ (the velocity-level constraint will hold regardless of the value of \ddot{C}_p^i if a sufficiently small integration step is taken) or (3) on the velocity level by directly requesting $\dot{C}_p^i \geq 0$. The constraint must be implemented on the velocity level if $\dot{C}_p^i < 0$. In practice, we make use of comparisons with tolerances when discriminating between the cases (1) – (3) and add extra terms to the velocity level and acceleration level formulations so that the original position level constraint would be stabilized, i.e. we use $J_{i,A_i} \cdot \vec{a}_{A_i} + J_{i,B_i} \cdot \vec{a}_{B_i} \geq -\vec{C}_p^i \cdot \alpha - J_i \cdot \vec{v} \cdot \beta - \dot{J}_i \cdot \vec{v}$ and $J_{i,A_i} \cdot \vec{v}_{A_i} + J_{i,B_i} \cdot \vec{v}_{B_i} \geq -\vec{C}_p^i \cdot \alpha$.

If $C_p^i < 0$ then the system state is invalid (ball penetrates the ground) and should be rejected. To handle this case, we would like to locate the state when $C_p^i = 0$ so that we could fall-back onto the previous case $C_p^i = 0$. For that, we can either (1) roll the simulation state back to the previous state and then use a bisection-like algorithm to locate the latest valid state $C_p^i = 0$ or (2) we can ignore the position error and act as if $C_p^i = 0$ (this way, we would only rely on the constraint stabilization mechanism to recover $C_p^i \geq 0$).

3.5 Contact

In this section we briefly discuss the problem of enforcing body non-penetration and modeling friction. We will account for these phenomena through constraints that will be associated with contacts reported by a collision detection library. Given the system state at time t , we will use the library to find which shapes are in contact and then formulate non-penetration and friction constraints specific to time t to constrain relative body motion at the contacting points. We informally define *contacts* as relevant coinciding points of contacting pairs of body shapes where contact forces or impulses should act in order to prevent penetration. We assume collision detection library describes contacts by vectors

$(A_i, B_i, \vec{p}_i, \vec{n}_i, \dot{\vec{n}}_i)$ such that contact i involves bodies A_i and B_i contacting at the world space point $\vec{p}_i = \vec{p}_{A_i} = \vec{p}_{B_i}$ (\vec{p}_{A_i} and \vec{p}_{B_i} are the corresponding points attached to A_i and B_i), the contact surface normal at \vec{p}_i is given by a unit vector \vec{n}_i pointing towards B_i and $\dot{\vec{n}}_i$ is the time derivative of \vec{n}_i . Let us denote $\vec{r}_{A_i} = \vec{p}_{A_i} - \vec{x}_{A_i}$ and $\vec{r}_{B_i} = \vec{p}_{B_i} - \vec{x}_{B_i}$.

We can then request *non-penetration at contact i* by stipulating that $C_i^n := \vec{n}_i \cdot (\vec{p}_{B_i} - \vec{p}_{A_i}) \geq 0$, which is a one-dimensional position level greater-or-equal constraint. Let's have a look at its time derivatives to retrieve the constraint's Jacobian blocks and the right-hand-side vector. We have $\dot{C}_i^n = \dot{\vec{n}}_i \cdot (\vec{p}_{B_i} - \vec{p}_{A_i}) + \vec{n}_i \cdot (\dot{\vec{p}}_{B_i} - \dot{\vec{p}}_{A_i})$, $\ddot{C}_i^n = \ddot{\vec{n}}_i \cdot (\vec{p}_{B_i} - \vec{p}_{A_i}) + \dot{\vec{n}}_i \cdot (\dot{\vec{p}}_{B_i} - \dot{\vec{p}}_{A_i}) + \vec{n}_i \cdot (\ddot{\vec{p}}_{B_i} - \ddot{\vec{p}}_{A_i})$. Because $\vec{p}_{A_i} = \vec{x}_{A_i} + \vec{r}_{A_i}$ and $\vec{p}_{B_i} = \vec{x}_{B_i} + \vec{r}_{B_i}$ we get $\dot{\vec{p}}_{A_i} = \dot{\vec{x}}_{A_i} + \vec{\omega}_{A_i} \times \vec{r}_{A_i}$, $\dot{\vec{p}}_{B_i} = \dot{\vec{x}}_{B_i} + \vec{\omega}_{B_i} \times \vec{r}_{B_i}$, $\ddot{\vec{p}}_{A_i} = \ddot{\vec{x}}_{A_i} + \dot{\vec{\omega}}_{A_i} \times \vec{r}_{A_i} + \vec{\omega}_{A_i} \times (\vec{\omega}_{A_i} \times \vec{r}_{A_i})$, $\ddot{\vec{p}}_{B_i} = \ddot{\vec{x}}_{B_i} + \dot{\vec{\omega}}_{B_i} \times \vec{r}_{B_i} + \vec{\omega}_{B_i} \times (\vec{\omega}_{B_i} \times \vec{r}_{B_i})$. Considering that $\vec{p}_{A_i} = \vec{p}_{B_i}$ at the contact, we then have $\dot{C}_i^n = \vec{n}_i \cdot (\dot{\vec{p}}_{B_i} - \dot{\vec{p}}_{A_i}) = \vec{n}_i \cdot \dot{\vec{x}}_{B_i} + \vec{n}_i \cdot (\vec{\omega}_{B_i} \times \vec{r}_{B_i}) - \vec{n}_i \cdot \dot{\vec{x}}_{A_i} - \vec{n}_i \cdot (\vec{\omega}_{A_i} \times \vec{r}_{A_i}) = \vec{n}_i \cdot \dot{\vec{x}}_{B_i} + (\vec{r}_{B_i} \times \vec{n}_i) \cdot \vec{\omega}_{B_i} - \vec{n}_i \cdot \dot{\vec{x}}_{A_i} - (\vec{r}_{A_i} \times \vec{n}_i) \cdot \vec{\omega}_{A_i}$ and $\ddot{C}_i^n = 2 \cdot \dot{\vec{n}}_i \cdot (\dot{\vec{p}}_{B_i} - \dot{\vec{p}}_{A_i}) + (\vec{n}_i \cdot \ddot{\vec{x}}_{B_i} + (\vec{r}_{B_i} \times \vec{n}_i) \cdot \dot{\vec{\omega}}_{B_i} - \vec{n}_i \cdot \ddot{\vec{x}}_{A_i} - (\vec{r}_{A_i} \times \vec{n}_i) \cdot \dot{\vec{\omega}}_{A_i} + \vec{n}_i \cdot (\vec{\omega}_{B_i} \times (\vec{\omega}_{B_i} \times \vec{r}_{B_i})) - \vec{n}_i \cdot (\vec{\omega}_{A_i} \times (\vec{\omega}_{A_i} \times \vec{r}_{A_i})))$. By comparing these equations against our constraint definitions,

we obtain $J_{i,A_i} = \begin{pmatrix} -\vec{n}_i^T & -(\vec{r}_{A_i} \times \vec{n}_i)^T \end{pmatrix}$, $J_{i,B_i} = \begin{pmatrix} \vec{n}_i^T & (\vec{r}_{B_i} \times \vec{n}_i)^T \end{pmatrix}$ and $J_i \cdot \vec{v}_{gen} = 2 \cdot \dot{\vec{n}}_i \cdot (\dot{\vec{x}}_{B_i} + \vec{\omega}_{B_i} \times \vec{r}_{B_i} - \dot{\vec{x}}_{A_i} - \vec{\omega}_{A_i} \times \vec{r}_{A_i}) + \vec{n}_i \cdot (\vec{\omega}_{B_i} \times (\vec{\omega}_{B_i} \times \vec{r}_{B_i}) - \vec{\omega}_{A_i} \times (\vec{\omega}_{A_i} \times \vec{r}_{A_i}))$.

The non-penetration constraint has the following direct interpretation. The value of C_i^n measures *relative body separation* at contact i , \dot{C}_i^n measures *normal velocity* at the contact and \ddot{C}_i^n measures *normal acceleration*. If λ_i^n is the Lagrange multiplier due to the acceleration level formulation of the constraint $\ddot{C}_i^n := J_i \cdot \vec{a}_{gen} - c_i \geq 0$ then λ_i^n equals the *normal force magnitude* and $-\vec{n}_i \cdot \lambda_i^n$ force is exerted on A_i at \vec{p}_{A_i} and $\vec{n}_i \cdot \lambda_i^n$ is exerted on B_i at \vec{p}_{B_i} . Our conditions from equation (6) due to non-penetration constraints thus say that both the normal acceleration and the normal force magnitude have to be non-negative and complementary to each other.

If λ_i^n is the Lagrange multiplier due to the velocity level formulation of the constraint $\dot{C}_i^n := J_i \cdot \vec{v}_{gen} \geq 0$ then $-\dot{C}_i^n$ measures the relative approaching body velocity and λ_i^n equals the normal impulse magnitude required to stop the bodies at \vec{p}_i so that $\dot{C}_i^n \geq 0$. This fact can be utilized to model *impacts*. Instead of stopping the bodies, we would like to take the velocity $-\dot{C}_i^n$ and have it opposed so that the bodies would bounce. This effect can be achieved by a minor modification of the original velocity level non-penetration constraint, (Baraff, 1997). Given a *coefficient of restitution* $\epsilon_i \in [0,1]$ at the contact i determining how bouncy the contacting surface is, we replace the non-penetration velocity level constraint $J_i \cdot \vec{v}_{gen} \geq 0$ with a constraint $J_i \cdot \vec{v}_{gen} \geq -\epsilon_i \cdot \dot{C}_i^n$.

Friction is usually modeled according to Coulomb friction law, (Trinkle et al., 1997). Coulomb law introduces friction forces acting along the contact's tangential plane. It extends the complementarity conditions on the normal acceleration and normal force magnitude by adding conditions on the direction and magnitude of the friction force, by relating the direction and magnitude to the relative tangential velocity at the contact and the corresponding normal force magnitude. The relation is quadratic and is most often linearized by considering two separate friction directions that the friction force can act along, (Trinkle et al., 1997).

In this linearization, friction at contact i is approximated by two additional *friction bounded equality constraints* that constrain relative body motion in two tangential directions perpendicular to each other and the contact normal. That is, if \vec{n}_i is the contact normal at contact i and \vec{t}_i^x and \vec{t}_i^y are two unit vectors such that $\vec{t}_i^x \cdot \vec{t}_i^y = 0$ and $\vec{t}_i^x \times \vec{t}_i^y = \vec{n}_i$ then it is requested that $C_i^x := \vec{t}_i^x \cdot (\vec{p}_{B_i} - \vec{p}_{A_i}) = 0$ and $C_i^y := \vec{t}_i^y \cdot (\vec{p}_{B_i} - \vec{p}_{A_i}) = 0$ subject to force limits. Friction constraints are special because their force limits are functions of the normal forces. If the corresponding non-penetration constraint $C_i^n \geq 0$ is implemented on the acceleration level then the friction constraints are implemented on the acceleration level as well by bounded equality constraints $\check{C}_i^x = 0$ and $\check{C}_i^y = 0$ with force limits $(\lambda_i^x)^{lo} = (\lambda_i^y)^{lo} = -\mu_i \cdot \lambda_i^n$ and $(\lambda_i^x)^{hi} = (\lambda_i^y)^{hi} = \mu_i \cdot \lambda_i^n$, where $\mu_i \geq 0$ is a static friction coefficient and λ_i^n is the normal force magnitude at contact i . Otherwise, the constraints are implemented on the velocity level by $\dot{C}_i^x = 0$ and $\dot{C}_i^y = 0$ with force limits $(\lambda_i^x)^{lo} = (\lambda_i^y)^{lo} = -\mu_i \cdot \lambda_i^n$ and $(\lambda_i^x)^{hi} = (\lambda_i^y)^{hi} = \mu_i \cdot \lambda_i^n$, where $\mu_i \geq 0$ is an impulsive friction coefficient and λ_i^n is the normal impulse magnitude at contact i , (Kawachi et al., 1997).

Note that for the friction constraints, force limits are not constant and depend on the values of λ_i^n . This implies we have to use a specialized solver to solve for $\vec{\lambda}$ or estimate λ_i^n first (e.g., we might ignore friction constraints and solve for an estimate of $\vec{\lambda}$, then fix the friction force limits using the estimated values of λ_i^n), (Smith, 2004). Friction constraints have analogous Jacobians and equation right-hand-side vectors as non-penetration constraints. It is just that \vec{n}_i vectors in the corresponding formulations are replaced with \vec{t}_i^x and \vec{t}_i^y vectors.

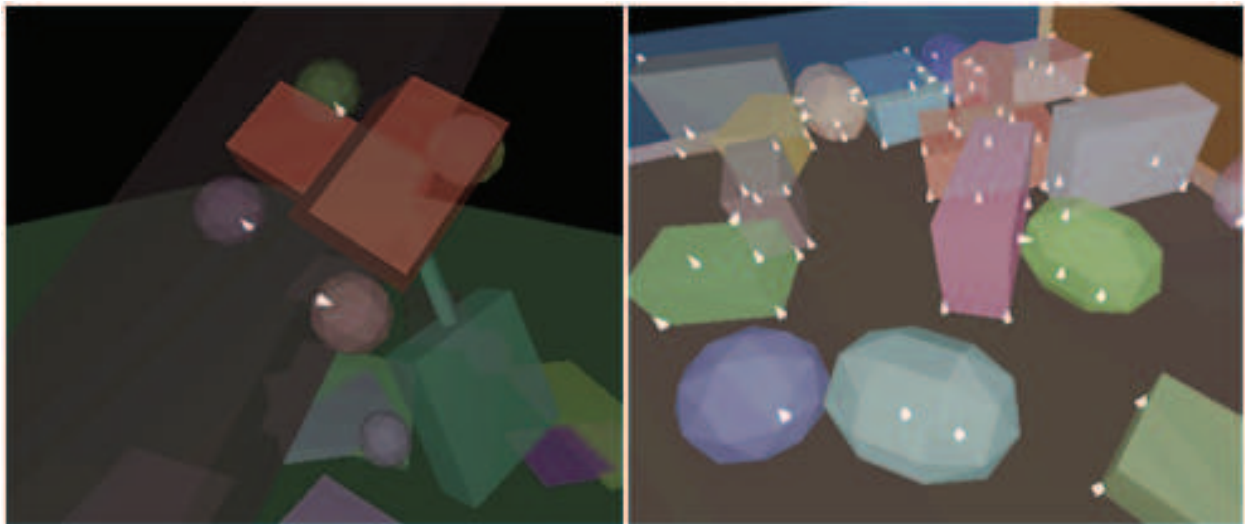


Fig. 4. Contact. Each contact point illustrated by a white spike yields a non-penetration acceleration constraint $\check{C}_i^n \geq 0$ and two friction bounded equality constraints $\check{C}_i^x = 0$ and $\check{C}_i^y = 0$.

4. Articulated bodies and motion control

In this section we will illustrate some examples of useful constraints that can be used to build articulated structures. In particular, we look at construction of joints (implemented with “anchors”) that connect individual rigid bodies by virtual hinges and restrict rotations about specific axes. Anchors are formally defined as position level equality constraints

(implemented either on the velocity or acceleration level) that restrict relative body positions and/or orientations. They implement various virtual joints (e.g., ball-and-socket or hinges) that connect bodies. This way, complex articulated jointed structures could be implemented.

4.1 Ball-and-socket joint

We start with a ball-and-socket joint that was discussed earlier (in the form of “point-to-point” constraints) in Section 3. The joint is defined in terms of two anchor points fixed in the coordinate frames of the first and the second constrained body. Ball-and-socket joint requires the anchors to occupy the same position in the world coordinate frame. As such, it removes 3 degrees of freedom from the system.

If i is the constraint index, A_i and B_i are the indices of the constrained bodies and $\vec{r}_{A_i}^b$ and $\vec{r}_{B_i}^b$ are the positions of the two anchors on the first and second body, expressed in the corresponding body-centric coordinate frames, then $\vec{p}_{A_i} = \vec{x}_{A_i} + R_{A_i} \cdot \vec{r}_{A_i}^b$ and $\vec{p}_{B_i} = \vec{x}_{B_i} + R_{B_i} \cdot \vec{r}_{B_i}^b$ are the world space positions of the anchors and the goal is to ensure that $\vec{p}_{A_i} = \vec{p}_{B_i}$. Let us denote $\vec{r}_{A_i} = \vec{p}_{A_i} - \vec{x}_{A_i}$ and $\vec{r}_{B_i} = \vec{p}_{B_i} - \vec{x}_{B_i}$. These vectors are fixed in the coordinate frame of the first and the second body respectively, resulting in $\dot{\vec{r}}_{A_i} = \vec{\omega}_{A_i} \times \vec{r}_{A_i}$ and $\dot{\vec{r}}_{B_i} = \vec{\omega}_{B_i} \times \vec{r}_{B_i}$. Following the derivation from the previous section, our constraint can then be formulated in the form of equations $\vec{C}_p^i := \vec{p}_{B_i} - \vec{p}_{A_i} = \vec{x}_{B_i} + \vec{r}_{B_i} - \vec{x}_{A_i} - \vec{r}_{A_i} = \vec{0}$, $\vec{C}_p^i := \vec{v}_{B_i} + \vec{\omega}_{B_i} \times \vec{r}_{B_i} - \vec{v}_{A_i} - \vec{\omega}_{A_i} \times \vec{r}_{A_i} = \vec{0}$ and $\vec{C}_p^i := \vec{a}_{B_i} - \vec{r}_{B_i}^* \cdot \vec{a}_{B_i} + \vec{\omega}_{B_i} \times (\vec{\omega}_{B_i} \times \vec{r}_{B_i}) - \vec{a}_{A_i} + \vec{r}_{A_i}^* \cdot \vec{a}_{A_i} - \vec{\omega}_{A_i} \times (\vec{\omega}_{A_i} \times \vec{r}_{A_i}) = \vec{0}$ and so we obtain $J_{i,A_i} = (-E \quad \vec{r}_{A_i}^*)$, $J_{i,B_i} = (E \quad -\vec{r}_{B_i}^*)$, $J_i \cdot \vec{v}_{gen} = \vec{\omega}_{B_i} \times (\vec{\omega}_{B_i} \times \vec{r}_{B_i}) - \vec{\omega}_{A_i} \times (\vec{\omega}_{A_i} \times \vec{r}_{A_i})$, $\vec{C}_p^i = \vec{p}_{B_i} - \vec{p}_{A_i}$, $m_i = 3$. These terms can be directly substituted into the acceleration level or velocity level formulations of the position level equality constraint.

4.2 Universal joint

Universal joint i attaches two bodies A_i and B_i like a point-to-point joint that additionally removes one rotational degree of freedom so that the constrained bodies can only rotate about two remaining axes. The two axes are defined explicitly and are perpendicular to one another. The first axis is attached to the first body A_i and the second axis is attached to the second body B_i .

The joint is defined by the positions of the two anchors $\vec{r}_{A_i}^b$ and $\vec{r}_{B_i}^b$ attached to the first and the second body and the directions of the two joint axes $\vec{u}_{A_i}^b$ and $\vec{u}_{B_i}^b$ attached to the first and the second body. Given the world space positions $\vec{p}_{A_i} = \vec{x}_{A_i} + R_{A_i} \cdot \vec{r}_{A_i}^b$ and $\vec{p}_{B_i} = \vec{x}_{B_i} + R_{B_i} \cdot \vec{r}_{B_i}^b$ of the anchors and the world space directions $\vec{u}_{A_i} = R_{A_i} \cdot \vec{u}_{A_i}^b$ and $\vec{u}_{B_i} = R_{B_i} \cdot \vec{u}_{B_i}^b$ of the axes, the constraint requires the anchors to occupy the same world space position $\vec{p}_{A_i} = \vec{p}_{B_i}$ and requests the axes to be perpendicular in world space such that $\vec{u}_{A_i} \cdot \vec{u}_{B_i} = 0$.

The constraint removes four degrees of freedom, $m_i = 4$, and is defined by four constraint rows $(\vec{C}_p^i)_k = 0$, $k = 1, \dots, 4$ of which the first three rows are due to the ball-and-socket joint discussed earlier and the fourth row is given by $(\vec{C}_p^i)_4 := -\vec{u}_{A_i} \cdot \vec{u}_{B_i} = 0$. The unstabilized velocity level formulation of the constraint is obtained by differentiating \vec{C}_p^i with respect to time. We have already seen this formulation for the first three rows and so only need to consider $(\dot{\vec{C}}_p^i)_4$. We have $(\dot{\vec{C}}_p^i)_4 := \frac{\partial}{\partial t}(-\vec{u}_{A_i} \cdot \vec{u}_{B_i}) = -(\vec{\omega}_{A_i} \times \vec{u}_{A_i}) \cdot \vec{u}_{B_i} - \vec{u}_{A_i} \cdot (\vec{\omega}_{B_i} \times \vec{u}_{B_i}) = \vec{\omega}_{B_i} \cdot (\vec{u}_{A_i} \times \vec{u}_{B_i}) - \vec{\omega}_{A_i} \cdot (\vec{u}_{A_i} \times \vec{u}_{B_i}) = 0$ and thus $(J_{i,A_i})_4 = \begin{pmatrix} 0 & -(\vec{u}_{A_i} \times \vec{u}_{B_i})^T \end{pmatrix}$ and $(J_{i,B_i})_4 =$

$\begin{pmatrix} 0 & (\vec{u}_{A_i} \times \vec{u}_{B_i})^T \end{pmatrix}$). Note that $\vec{\omega}_{B_i} \cdot (\vec{u}_{A_i} \times \vec{u}_{B_i}) - \vec{\omega}_{A_i} \cdot (\vec{u}_{A_i} \times \vec{u}_{B_i}) = (\vec{u}_{A_i} \times \vec{u}_{B_i}) \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i})$ equals the relative angular velocity (rotation speed) about the axis $\vec{u}_{A_i} \times \vec{u}_{B_i}$ and so the velocity-level constraint $(\dot{\vec{C}}_p^i)_4 = 0$ explicitly prohibits relative body rotation about $\vec{u}_{A_i} \times \vec{u}_{B_i}$.

In turn, the two constrained bodies can only rotate about \vec{u}_{A_i} and \vec{u}_{B_i} .

The unstabilized acceleration level formulation of the constraint is obtained by differentiating the velocity level formulation with respect to time. The first three rows of the unstabilized acceleration constraint are again the same as in the ball-and-socket joint case and the fourth row is given by $(\ddot{\vec{C}}_p^i)_4 := \frac{\partial}{\partial t}((\vec{u}_{A_i} \times \vec{u}_{B_i}) \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i})) = \left(\frac{\partial \vec{u}_{A_i}}{\partial t} \times \vec{u}_{B_i} + \vec{u}_{A_i} \times \frac{\partial \vec{u}_{B_i}}{\partial t}\right) \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i}) + (\vec{u}_{A_i} \times \vec{u}_{B_i}) \cdot (\ddot{\alpha}_{B_i} - \ddot{\alpha}_{A_i}) = ((\vec{\omega}_{A_i} \times \vec{u}_{A_i}) \times \vec{u}_{B_i} + \vec{u}_{A_i} \times (\vec{\omega}_{B_i} \times \vec{u}_{B_i})) \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i}) + (\vec{u}_{A_i} \times \vec{u}_{B_i}) \cdot (\ddot{\alpha}_{B_i} - \ddot{\alpha}_{A_i}) = 0$ because \vec{u}_{A_i} is attached to the first body A_i and \vec{u}_{B_i} is attached to the second body B_i . We thus obtain $(\vec{J}_i \cdot \vec{v}_{gen})_4 = ((\vec{\omega}_{A_i} \times \vec{u}_{A_i}) \times \vec{u}_{B_i} + \vec{u}_{A_i} \times (\vec{\omega}_{B_i} \times \vec{u}_{B_i})) \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i})$.

4.3 Hinge joint

Hinge joint i attaches two bodies like a ball-and-socket joint but additionally removes two more rotational degrees of freedom so that the constrained bodies A_i and B_i can only rotate about a single common axis (hinge axis). The joint is defined by the positions of the two anchors $\vec{r}_{A_i}^b$ and $\vec{r}_{B_i}^b$ attached to the first and the second body and the directions of the axes $\vec{u}_{A_i}^b$ and $\vec{u}_{B_i}^b$ attached to the first and the second body. Given the world space positions $\vec{p}_{A_i} = \vec{x}_{A_i} + R_{A_i} \cdot \vec{r}_{A_i}^b$ and $\vec{p}_{B_i} = \vec{x}_{B_i} + R_{B_i} \cdot \vec{r}_{B_i}^b$ of the anchors and the world space directions $\vec{u}_{A_i} = R_{A_i} \cdot \vec{u}_{A_i}^b$ and $\vec{u}_{B_i} = R_{B_i} \cdot \vec{u}_{B_i}^b$ of the axes, the constraint requires the anchors to occupy the same position in the world coordinate frame $\vec{p}_{A_i} = \vec{p}_{B_i}$ and requests the hinge axes to align in world space such that $\vec{u}_{A_i} = \vec{u}_{B_i}$.

This time, we will show the velocity level formulation of the constraint directly because such a formulation allows to naturally express conditions on what rotation axes the constrained bodies cannot rotate about. The constraint removes 5 degrees of freedom, $m_i = 5$, and is defined by five velocity constraint rows $(\dot{\vec{C}}_p^i)_k = 0$, $k = 1, \dots, 5$ of which the first three rows are due to the point-to-point joint discussed earlier. We will now define the remaining two constraint rows $(\dot{\vec{C}}_p^i)_4 = 0$ and $(\dot{\vec{C}}_p^i)_5 = 0$.

Let \vec{g} and \vec{h} be two world space vectors perpendicular to \vec{u}_{A_i} and assume that $\vec{u}_{A_i} = \vec{u}_{B_i}$. To make sure that the two bodies can rotate only about \vec{u}_{A_i} , the relative angular velocity $(\vec{\omega}_{B_i} - \vec{\omega}_{A_i}) \cdot \vec{w}$ about any axis \vec{w} perpendicular to \vec{u}_{A_i} (relative rotation speed about \vec{w}) must be zero. This can be enforced by simply requiring that the relative angular velocity about \vec{g} and \vec{h} is zero. We thus get $(\dot{\vec{C}}_p^i)_4 := \vec{g} \cdot \vec{\omega}_{B_i} - \vec{g} \cdot \vec{\omega}_{A_i} = 0$ and $(\dot{\vec{C}}_p^i)_5 := \vec{h} \cdot \vec{\omega}_{B_i} - \vec{h} \cdot \vec{\omega}_{A_i} = 0$ and so can write $(J_{i,A_i})_4 = (\vec{0}^T \quad -\vec{g}^T)$, $(J_{i,B_i})_4 = (\vec{0}^T \quad \vec{g}^T)$ and $(J_{i,A_i})_5 = (\vec{0}^T \quad -\vec{h}^T)$, $(J_{i,B_i})_5 = (\vec{0}^T \quad \vec{h}^T)$.

Now, if the constraint is broken and the hinge axes are not aligned, we can express the orientation error by a vector $\vec{u}_{B_i} \times \vec{u}_{A_i}$. To stabilize the constraint and eventually bring the axes back into alignment, we want to rotate the bodies about $\vec{z} = \frac{\vec{u}_{B_i} \times \vec{u}_{A_i}}{\|\vec{u}_{B_i} \times \vec{u}_{A_i}\|}$ with a speed

proportional to $\|\vec{u}_{B_i} \times \vec{u}_{A_i}\|$. Because \vec{z} lies in the plane perpendicular to \vec{u}_{A_i} , the stabilization can be decomposed to requests on rotation speeds about \vec{g} and \vec{h} and thus directly incorporated into the constraint equations as $(\dot{\vec{C}}_p^i)_4 := \vec{g} \cdot \vec{\omega}_{B_i} - \vec{g} \cdot \vec{\omega}_{A_i} - \vec{g} \cdot (\vec{u}_{B_i} \times \vec{u}_{A_i}) \cdot \alpha = 0$ and $(\dot{\vec{C}}_p^i)_5 := \vec{h} \cdot \vec{\omega}_{B_i} - \vec{h} \cdot \vec{\omega}_{A_i} - \vec{h} \cdot (\vec{u}_{B_i} \times \vec{u}_{A_i}) \cdot \alpha = 0$, where α is the stabilization constant. We can thus write $(\vec{C}_p^i)_4 \approx \vec{g} \cdot (\vec{u}_{A_i} \times \vec{u}_{B_i})$ and $(\vec{C}_p^i)_5 \approx \vec{h} \cdot (\vec{u}_{A_i} \times \vec{u}_{B_i})$.

The unstabilized acceleration level formulation of the constraint is obtained by differentiating the velocity level formulation with respect to time. The first three rows of the unstabilized acceleration constraint are the same like in the point-to-point case and the remaining two rows are given by $(\ddot{\vec{C}}_p^i)_4 := (\vec{\omega}_{A_i} \times \vec{g}) \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i}) + \vec{g} \cdot (\vec{\alpha}_{B_i} - \vec{\alpha}_{A_i}) = 0$ and $(\ddot{\vec{C}}_p^i)_5 := (\vec{\omega}_{A_i} \times \vec{h}) \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i}) + \vec{h} \cdot (\vec{\alpha}_{B_i} - \vec{\alpha}_{A_i}) = 0$ because \vec{g} and \vec{h} can be assumed to be attached to the first body A_i . We thus obtain $(\vec{J}_i \cdot \vec{v}_{gen})_4 = (\vec{\omega}_{A_i} \times \vec{g}) \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i})$ and $(\vec{J}_i \cdot \vec{v}_{gen})_5 = (\vec{\omega}_{A_i} \times \vec{h}) \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i})$. By now we have defined all the terms J_{i,A_i} , J_{i,B_i} , $\vec{J}_i \cdot \vec{v}_{gen}$, \vec{C}_p^i and m_i required to implement hinge joint either on the velocity or acceleration level with stabilization.

5. Motion control and motors

In this section we show how to actuate the articulated body defined in the previous section. In particular, we show how constraints can be used to define motors that can be applied at the joints of articulated body, similarly to (Kokkevis, 2004). Motors control angles or displacements along certain axes; they also control rotational or translational speed.

Motors provide mechanisms to actuate body pairs by controlling their relative linear and angular properties. Given desired relative positions, orientations or velocities that need be followed by a motor, the motor formulates appropriate position level or velocity level constraints. These constraints are then implemented on the velocity or acceleration level by submitting corresponding (bounded) equality or inequality constraints with stabilization. Most often, motors are combined with anchors so that bodies can be actuated about the valid (originally unconstrained) degrees of freedom at joints, like a hinge axis or universal joint's axes.

Anchors discussed earlier were implemented by equality constraints that were always "active". Once constrained bodies are connected by a joint, the corresponding anchor constraint becomes effective and is incrementally maintained. With motors, the corresponding constraint changes in accordance with desired motion. The motor constraint is not permanent and the system state is often inconsistent with the constraint formulation when the constraint is requested to be followed. As a result, the constraint has to be stabilized and it is the constraint stabilization mechanism and/or impulses due to first-order dynamics that generate the energy needed to actuate the bodies. For example, when a position motor, following a desired position of the second body relative to the position of the first body, is implemented on the acceleration level by submitting an acceleration constraint, both constraint position and velocity errors have to be stabilized and the necessary actuation energy is generated by the constraint's stabilization mechanism. Stabilization parameters determine how long it takes the motor to approach the desired position.

We will now discuss various types of motors. Each particular type will define a motor-specific measure of relative body state relating the state of one actuated body to the state of the other actuated body. Then, given the desired (intended) value and the current value of this measure (e.g., relative angle, relative orientation, relative position in space, relative rotation speed, relative angular velocity, relative linear velocity), the motor formulates motion control constraints that would drive the measure towards its desired value.

We distinguish between linear and angular motors. Pure linear motors affect only linear properties of the bodies while pure angular motors affect only angular properties. As a convention, we assume that the measure used by motor i actuating bodies A_i and B_i always relates the state of the second body B_i to the state of the first body A_i . In addition, for the sake of convenience, we assume that the position level measure is defined with respect to the initial state at time t_0 (in other words the position measure is always equal to a zero vector at t_0).

5.1 3-DOF linear motor

We start with a simple *3-DOF velocity level linear motor i* that controls relative linear velocity of A_i and B_i along world space axes. The motor velocity measure $\dot{\vec{u}}(t) := \vec{v}_{B_i}(t) - \vec{v}_{A_i}(t)$ is equivalent to relative body velocity in the world coordinate frame and is independent of relative angular properties. Then, given a desired value $\dot{\vec{u}}_d(t)$ of this measure, we impose a three dimensional velocity level constraint, $m_i = 3$, $\vec{C}_v^i := \dot{\vec{u}}(t) - \dot{\vec{u}}_d(t) = \vec{0}$. This constraint affects only linear properties of the two bodies and can be implemented either directly on the velocity level or the acceleration level by requesting $\vec{C}_v^i := \vec{a}_{B_i}(t) - \vec{a}_{A_i}(t) = -\vec{C}_v^i \cdot \beta$, where β is a velocity stabilization parameter. The formulations produce $J_{i,A_i} = (-E \ 0)$, $J_{i,B_i} = (E \ 0)$ and $\dot{J}_i \cdot \vec{v}_{gen} = 0$.

Now, if we are to implement a *3-DOF position level linear motor i* that controls relative position of two bodies A_i and B_i in the world coordinate frame, we can define the motor position measure $\vec{u}(t)$ as a vector of current relative body displacements along the world space axes, that is, $\vec{u}(t) := (\vec{x}_{B_i}(t) - \vec{x}_{A_i}(t)) - (\vec{x}_{B_i}(t_0) - \vec{x}_{A_i}(t_0))$ and then, given a desired value $\vec{u}_d(t)$ of the measure, we can impose a three dimensional position level constraint, $m_i = 3$, $\vec{C}_p^i(t) := \vec{u}(t) - \vec{u}_d(t) = \vec{0}$. We get $\vec{C}_p^i = \dot{\vec{u}}(t) - \dot{\vec{u}}_d(t) \approx \dot{\vec{u}}(t) = \vec{v}_{B_i}(t) - \vec{v}_{A_i}(t)$ and can once again implement the position level constraint on the velocity level by requesting $\vec{C}_v^i := \vec{v}_{B_i}(t) - \vec{v}_{A_i}(t) = -\vec{C}_p^i \cdot \alpha$, where α is a position stabilization constant. This lets us implement the position level linear motor by using the corresponding velocity level linear motor.

5.2 3-DOF angular motor

Let's first present a *3-DOF velocity level angular motor i* that directly controls relative angular velocity of A_i and B_i in the world coordinate frame, both in terms of the rotation direction and speed. This motor is an angular analog to the 3-DOF linear velocity motor presented earlier and hence we can simply define the three dimensional motor velocity measure $\dot{\vec{u}}(t)$ as $\dot{\vec{u}}(t) := \vec{\omega}_{B_i}(t) - \vec{\omega}_{A_i}(t)$. Then, given a desired value $\dot{\vec{u}}_d(t)$ of the velocity, we impose a three dimensional velocity level constraint, $m_i = 3$, $\vec{C}_v^i := \dot{\vec{u}}(t) - \dot{\vec{u}}_d(t) = \vec{0}$ that can be implemented either directly on the velocity level or the acceleration level by requesting

$\dot{\vec{C}}_v^i = -\vec{C}_v^i \cdot \beta$, where β is a velocity stabilization parameter. We then get $J_{i,A_i} = \begin{pmatrix} 0 & -E \end{pmatrix}$, $J_{i,B_i} = \begin{pmatrix} 0 & E \end{pmatrix}$ and $\dot{J}_i \cdot \vec{v}_{gen} = 0$.

We now present a 3-DOF position level angular motor i , implemented according to (Vondrak, 2006), that controls relative orientation of the two bodies. The motor constraint will be formulated directly on the velocity level. This simplifies the formulation but still allows us to directly specify the world space axis that the actuated bodies have to rotate about and how fast in order to reach the desired relative orientation. We define orientation of B_i relative to the orientation of A_i at time t as $R_{A_i}^{-1}(t) \cdot R_{B_i}(t)$.

To formulate the constraint, we first need to define the motor measure, relative orientation, $U(t)$. Following our convention, $U(t)$ measures orientation of B_i relative to A_i relative to the initial relative orientation $R_{A_i}^{-1}(t_0) \cdot R_{B_i}(t_0)$ and hence $U(t)$ is defined as $U(t) := R_{A_i}^{-1}(t) \cdot R_{B_i}(t) \cdot R_{B_i}^{-1}(t_0) \cdot R_{A_i}(t_0)$. When we are given a desired value $U_d(t)$ for the relative orientation, we need to find the world space rotation axis \vec{w} and an angle z that the two bodies have to rotate about so that $U = U_d$ after the rotation is applied. For that, we define relative B_i to A_i rotational error $U_e(t) := R_{A_i}^{-1}(t) \cdot R_{B_i}(t) \cdot R_{B_i}^{-1}(t_0) \cdot R_{A_i}(t_0) \cdot U_d^{-1}(t)$ which we then decompose to a rotation axis $\vec{w}_{A_i}^b$ (attached to A_i) and an angle $z \in [0, \pi)$. We then transform $\vec{w}_{A_i}^b$ to world coordinate frame obtaining the axis $\vec{w} = R_{A_i}(t) \cdot \vec{w}_{A_i}^b$. Because the relative rotation about \vec{w} by z is the only valid relative rotation consistent with our orientation request, we can finally constrain relative body angular velocity by stipulating that $\vec{C}_v^i := \vec{\omega}_{B_i} - \vec{\omega}_{A_i} = -\vec{w} \cdot z \cdot \alpha$, where α is a position stabilization constant. This lets us implement the position level angular motor by using the corresponding velocity level angular motor that follows $\vec{u}_d(t) = -\vec{w} \cdot z \cdot \alpha$.

5.3 1-DOF motors

We will now present a template for 1-DOF linear and/or angular position and velocity motors that control relative displacement or angle (generally called the offset) and their time derivatives along or about given fixed axes attached to the first body in the actuated body pair. Unlike the previous cases, our 1-DOF motor will not define its position-level motor measure $u(t)$, the offset of B_i from A_i relative to the initial offset at time t_0 , explicitly. In contrast, it will be assumed that at any time t the value of $u(t)$ specific to the motor could be determined. That is because 1-DOF motors will be most often combined with anchors and the value of $u(t)$ will be specific to the anchor type. For example, the hinge joint's hinge axis could be controlled by a 1-DOF motor and the value of $u(t)$ in that case would equal the rotation angle of B_i about the hinge axis relative to the initial relative orientation of B_i and A_i . The motors will additionally provide support for enforcing one-dimensional joint displacement and angle limits because the enforcement in the 1-DOF case can be implemented by using simple one-dimensional constraints, given the value of $u(t)$.

5.3.1 1-DOF angular motor

Let's consider the 1-DOF angular position and velocity motor i that actuates bodies A_i and B_i and is specified by a rotation axis $\vec{w}_{A_i}^b$ attached to the first body A_i , motor angle measure $u(t)$ and motor angle limits z_i^{lo} and z_i^{hi} . Given the world space rotation axis $\vec{w}_i = R_{A_i} \cdot \vec{w}_{A_i}^b$, the value of the angle measure $u(t)$ is supposed to equal the angle the second body B_i is rotated about \vec{w}_i with respect to the first body A_i , relative to the initial relative body orientation. We

then define velocity measure $\dot{u}(t)$ as the time derivative of the angle measure $u(t)$. The velocity measure equals the relative rotation speed about \vec{w}_i and so is given by $\dot{u}(t) = \vec{w}_i \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i})$. We will say the motor's axis is limited, if the angle limits are in effect and we will say the axis is powered if the motor is to follow either a desired angle $u_d(t)$ or a desired rotation speed $\dot{u}_d(t)$.

We now show velocity level constraints with stabilization implementing the motor. The constraints ensure that angle limits are obeyed and a desired angle or angular velocity is followed. If the axis is limited and $u(t) \leq z_i^{lo}$ then it is requested that $C_v^i := \vec{w}_i \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i}) - (z_i^{lo} - u(t)) \cdot \alpha \geq 0$, where α is a position level stabilization constant. If the axis is limited and $u(t) \geq z_i^{hi}$ then $C_v^i := \vec{w}_i \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i}) - (z_i^{hi} - u(t)) \cdot \alpha \leq 0$ is requested. If the motor is to follow a desired angle $u_d(t)$ then it is requested that $C_v^i := \vec{w}_i \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i}) - (u_d(t) - u(t)) \cdot \alpha = 0$. If the motor is to follow a desired angular velocity $\dot{u}_d(t)$ then we request $C_v^i := \vec{w}_i \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i}) - \dot{u}_d(t) = 0$. Each constraint produces the same Jacobian blocks $J_{i,A_i} = (\vec{0}^T \quad -\vec{w}_i^T)$ and $J_{i,B_i} = (\vec{0}^T \quad \vec{w}_i^T)$. By differentiation, we obtain $\dot{J}_i \cdot \vec{v}_{gen} = (\vec{\omega}_{A_i} \times \vec{w}_i) \cdot (\vec{\omega}_{B_i} - \vec{\omega}_{A_i})$.

5.3.2 1-DOF linear motor

To formulate a 1-DOF linear position and velocity motor i we proceed similarly to the 1-DOF angular motor case. The motor is specified by a translation axis $\vec{w}_{A_i}^b$ attached to the center of mass of A_i , motor displacement measure $u(t)$ of the second body's center of mass from the first body's center of mass along \vec{w}_i (relative to the initial displacement) and motor displacement limits z_i^{lo} and z_i^{hi} . Because $\vec{w}_{A_i}^b$ is attached to A_i , the relative displacement $u(t)$ is a function of $R_{A_i}(t)$ and so both linear and angular properties have to be constrained. Here, we constrain the linear properties only and assume the associated anchor will impose additional constraints to maintain proper body orientation, if necessary. This lets us formulate the same set of constraints as in the previous section, but this time, linear velocities are constrained instead of angular velocities. We get $J_{i,A_i} = (-\vec{w}_i^T \quad \vec{0}^T)$, $J_{i,B_i} = (\vec{w}_i^T \quad \vec{0}^T)$ and $\dot{J}_i \cdot \vec{v}_{gen} = (\vec{\omega}_{A_i} \times \vec{w}_i) \cdot (\vec{v}_{B_i} - \vec{v}_{A_i})$.

5.4 Trajectory motor

When controlling motion of loop-free articulated structure it is sometimes convenient to specify desired position of selected points (markers), rigidly attached to the segments, instead of specifying desired angles at joints. Given the desired positions of markers, one could use inverse kinematics to solve for desired angles. These desired angles can then be submitted as constraints for motors, resulting in the motion of the body that would implicitly satisfy desired positions of the markers.

We propose to use a different approach, previously used in (Vondrak et al., 2008), based on first-order inverse dynamics – given the desired positions of markers, we solve for desired angular velocities that should be followed by the structure's angular motors so that markers would reach their desired positions. Using inverse dynamics is simpler in our case, because we already formulated how to solve for velocities that satisfy a given set of constraints (by solving for the impulses).

Let's assume our articulated structure is made up of n rigid body segments connected by $n - 1$ actuated joints of various kinds. We assume the root segment r has index 1 and that

the segment $i = 2, \dots, n$ is connected to its parent segment by joint i . In addition, let's connect the root segment $r = 1$ to an external fixed body (i.e., the world, implemented as a rigid body with infinite mass) by an additional 6-DOF motor r that is a combination of the 3-DOF linear and the 3-DOF angular motors presented earlier. The external fixed body establishes a coordinate frame in world space and so the position measure $\vec{q}_r \in \mathbf{R}^6$ of the combined 6-DOF motor defines the global position and orientation of the structure's root segment in world space. The other motor measures \vec{q}_i , $i = 2, \dots, n$, recursively parameterize the positions and orientations of the remaining segments in world space and so we can define the kinematic state \vec{q} of the structure as a vector $\vec{q} = (\vec{q}_1, \vec{q}_2, \dots, \vec{q}_n)$. We also define the structure's velocity $\dot{\vec{q}} = (\dot{\vec{q}}_1, \dot{\vec{q}}_2, \dots, \dot{\vec{q}}_n)$ which is a concatenation of the structure's motor velocity measures.

Now, if there are m markers $j = 1, \dots, m$ attached to the kinematic structure such that the world space position of marker j at state \vec{q} is given by $\vec{z}^j(\vec{q})$ and its corresponding desired world space position is given by \vec{z}_d^j , the inverse kinematics approach would solve for desired position measures \vec{q}_d so that (1) $\vec{z}^j(\vec{q}_d) = \vec{z}_d^j$ for all $j = 1, \dots, m$ and (2) $(\vec{q}_d)_1 = \vec{q}_1$. The first constraint requests the markers to reach their desired positions at state at \vec{q}_d while the second constraint fixes the position and orientation of the root segment to ensure that requests (1) cannot be satisfied by simple translation or rotation of the root segment. Once \vec{q}_d values are known, structure's motors would be programmed to follow \vec{q}_d .

With the inverse dynamics approach, we want to solve for desired velocity measures $\dot{\vec{q}}_d$ so that (1) $\dot{\vec{z}}^j = -(\vec{z}^j(\vec{q}) - \vec{z}_d^j) \cdot \alpha$, for all $j = 1, \dots, m$, where α is a position stabilization constant and (2) $(\dot{\vec{q}}_d)_1 = \dot{\vec{q}}_1$. The first constraint requests the markers to be moved towards their desired positions while the second constraint ensures that constraints (1) cannot be solved by forced translation or rotation of the root segment. Because each equation (1) is actually a constraint on relative velocities of two points (anchors), in practice, the value of $\dot{\vec{q}}_d$ can be computed by (1) attaching m ball-and-socket joints between anchors $\vec{z}^j(\vec{q})$ on the structure parts and anchors \vec{z}_d^j on the world body, (2) programming the root segment's 6-DOF motor to follow the current values of the velocity measures $\dot{\vec{q}}_1$ and taking an auxiliary simulation step from the current time t to a time $t + \Delta t$ to retrieve the figure velocities $\dot{\vec{q}}(t + \Delta t)$ (if we need to enforce additional constraints, e.g. due to joint angle limits or angle-based actuation of other joints, we can just let those constraints be active at this time). These velocities can then be used as the desired velocities $\dot{\vec{q}}_d$. We can thus let $\dot{\vec{q}}_d := \dot{\vec{q}}(t + \Delta t)$. Once the $\dot{\vec{q}}_d$ values are known, we roll the simulation state back to t and reprogram the structure's motors to follow $\dot{\vec{q}}_d$.

5.5 Control example

In this section, we illustrate the use of joint and motor constraints to build a simple humanoid character that will be animated using a physical simulation. Motion control constraints will be set up such that desired motion specified by given motion capture data will be followed. We assume that the desired motion is physically valid.

We compose the character out of rigid bodies corresponding to the character's body parts (in lieu of Section 4) and then connect these parts by actuated ball-and-socket joints that have 3-DOF angular motors associated with them so that the character's motion could be controlled. We use 3-DOF linear and 3-DOF angular motors to directly control the position

and orientation of the character's root segment in world space. The direct control of the root segment (while not entirely physically plausible) makes control simple, such that we don't have to worry about character's balance and/or unanticipated collisions with the environment. In addition, to allow the motion of the character to adjust to external disturbances, such as dragging of body parts by a mouse, in a plausible way, we set the angular motors actuating the character's upper body, head and arms to generate bounded constraint forces³. To produce the final animation, we program the character's motors to follow desired positions and joint orientations given by motion capture and simulate the rigid body system forward in time.

Figure 5 shows animation results generated by (Vondrak, 2006) using the described character model. The top row shows the animation when no external disturbances are present. The middle row illustrates effects of attaching a basket to the character's right hand by a hinge joint. The axis motor is programmed to maintain zero rotation speed subject to force limits so that friction between the hand and the basket handle could be modeled. The bottom row extends the previous case by replacing the basket with an umbrella. The umbrella is pulled to the front of the character's head by a ball-and-socket joint attached to the right wrist and an anchor in front of the head. In addition, two 1-DOF angular motors are attached to the umbrella and the external world body to keep the umbrella's "pitch" and "bank" angles close to zero so that the umbrella will be held upright regardless of the character's pose.

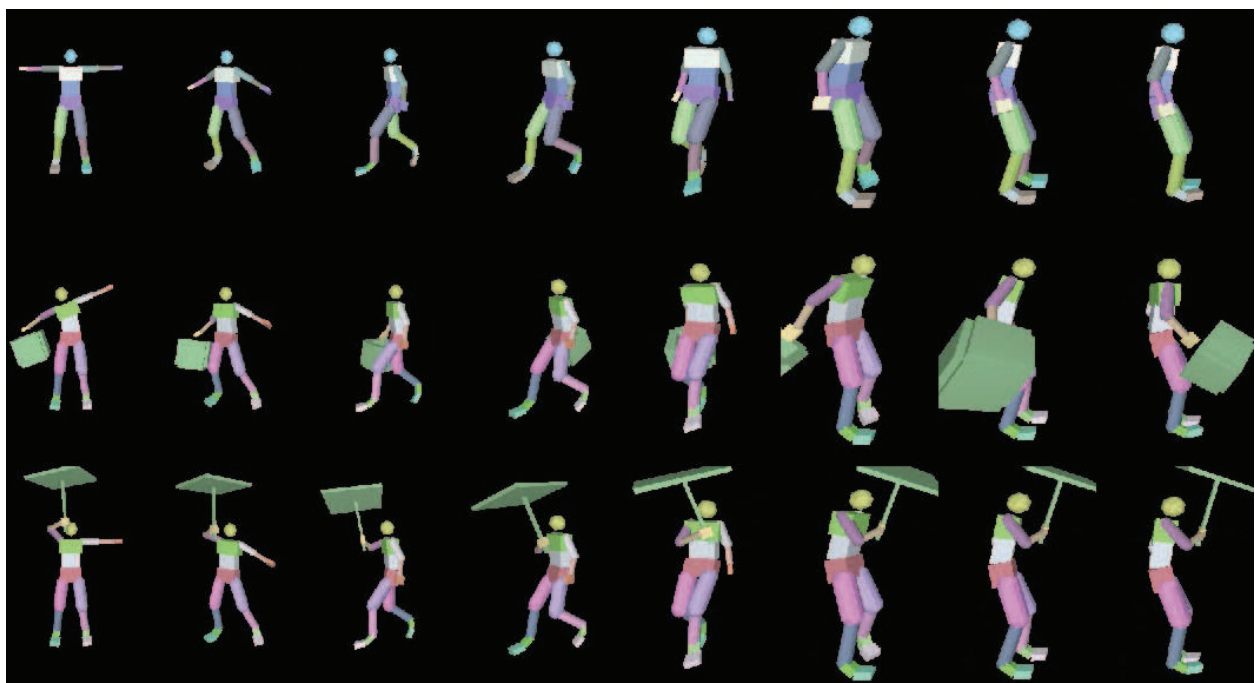


Fig. 5. Physics-based animation of an articulated character under the effects of different external disturbances. Top row shows the original motion of a subject preparing to make a jump when no external disturbances are present, the other rows show adjusted motion when objects are attached to the character's right hand.

³ Motors actuating lower body and legs are programmed to generate almost unbounded forces so that the direct control of the character's (root) position and orientation is consistent with the actual motion of the lower body.

6. Discussion and conclusions

Physics-based simulation has emerged as a popular approach for realistic animation and analysis of rigid and articulated bodies in motion. This chapter briefly reviewed basic principles of unconstrained rigid body mechanics and then focused on the more challenging constrained rigid body mechanics principles. We have outlined basic concepts of rigid and articulated body modeling and simulation and advocated a constraint-based motion control that is based on motors implemented by constraints imposed on the position, velocity and/or acceleration of joint angles or points rigidly attached to the bodies.

The formulated approach to control is simple and accurate within the context of readily available physics-based engines. That said, general control of complex articulated models, such as humanoids, is very challenging, especially in absence of trajectories that constrain all or most parts of the body over time. In particular, design of controllers that reproduce dynamics and energetics of human motion as well as can model dynamic variations due to the physical morphology or style of the individual remains an open issue. A variety of other approaches to motion control exist (see Section 1.1). For instance, task-based control (where the user specifies the task instead of joint angles or trajectories, e.g., pick up a mug from the table) has been emerging as the new alternative direction in the control and has a number of appealing properties from the point of view of animators and game designers. Discussing these alternative approaches falls outside the scope of this chapter. Lastly, we also do not consider numerical and performance aspects of the constraint-based motion control method and do not discuss various integration methods that clearly affect the quality (and the speed) of resulting simulations (see (Boeing et al., 2007) for discussion).

That said, constraint-based motion control has become the standard approach for animating virtual worlds with stunning realism. This approach is versatile enough to model distinct phenomena like body articulation, joint actuation and contact in a uniform way; it is also capable of producing stable high quality simulations with predictable results in real time. Consequently, constraint-based control has become the default motion control strategy employed by all major commercial and open-source simulation packages.

7. References

- Abe, Y.; Popovic, J. (2006). Interactive animation of dynamic manipulation. In: Eurographics/ACM SIGGRAPH Symposium on Computer Animation
- Baraff, D. (1996). Linear-Time Dynamics using Lagrange Multipliers, In: SIGGRAPH'96, pages 137 – 146, New Orleans
- Baraff, D.; Witkin, A. & Kass M. (1997). An Introduction to Physically-Based Modeling, In: SIGGRAPH'97. Course Notes
- Bourg, D. (2002). Physics for Game Developers, O'Reilly & Associates, Inc.
- Boeing, A. & Braunl, T (2007). Evaluation of real-time physics simulation systems. In: International Conference on Computer Graphics and Interactive Techniques (Graphite), pages 281 – 288
- Cline, M. (2002). Rigid Body Simulation with Contact and Constraints, In: Master's Thesis, The University of British Columbia
- Eberly, D. (2003). Game Physics (Interactive 3d technology series), MorganKaufman
- Erleben, K. (2002). Rigid Body Simulation, In: Lecture Notes
- <http://www.diku.dk/forskning/image/teaching/Courses/e02/RigidBodySimulation/>

- Faloutsos, P.; van de Panne, M. & Terzopoulos, D. (2001). Composable controllers for physics-based character animation. In: ACM Transactions on Computer Graphics (SIGGRAPH)
- Goswami, A.; Espiau, B. & Keramane, A (1996). Limb cycles and their stability in a passive bipedal gait, In: IEEE Robotics and Automation
- Havok Inc. (2007). Havok behavior. <http://www.havok.com/>
- Hodgins, J.; Wooten, W.; Brogan, D. & O'Brien, J. (1995). Animating human athletics. In: ACM Transactions on Computer Graphics (SIGGRAPH)
- Kawachi, K.; Suzuki, H. & Kimura, F. (1997). Simulation of Rigid Body Motion with Impulsive Friction Force, In: Proceedings of IEEE International Symposium on Assembly and Task Planning, pages 182 - 187
- Khatib, O. (1987). A unified approach to motion and force control of robot manipulators: The operational space formulation, In: IEEE Journal on Robotics and Automation 3, 1 (February), pages 43-55
- Kokkevis, E. (2004). Practical Physics for Articulated Characters
- Laszlo, J. F.; van de Panne, M. & Fiume, E. (1996). Limit Cycle Control and its Application to the Animation of Balancing and Walking, In: ACM Transactions on Computer Graphics (SIGGRAPH)
- Liu, C. K.; Hertzmann, A. & Popovic, Z. (2005). Learning physics-based motion style with nonlinear inverse optimization, In: ACM Transactions on Graphics, 24(3):1071.1081
- McCann, J.; Pollard, N. S. & Srinivasa, S. (2006). Physics-Based Motion Retiming, In: ACM Transactions on Computer Graphics (SIGGRAPH)/Eurographics Symposium on Computer Animation
- Nakamura, Y.; Hhanafusa, H. & Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators, In: International Journal of Robotics Research, 6(2), pages 3-15
- PhysX. NVIDIA. http://www.nvidia.com/object/physx_new.html
- Safonova A.; Hodgins J. K. & Pollard N. S. (2004). Synthesizing Physically Realistic Human Motion in Low-Dimensional, Behavior-Specific Spaces, In: ACM Transactions on Graphics 23(3), SIGGRAPH
- Shapiro, A.; Chu, D.; Allen, B. & Faloutsos, P. (2007). A Dynamic Controller Toolkit, In: ACM Transactions on Graphics Video Game Symposium (Sandbox)
- Smith, R. (2004). Open Dynamics Engine, In: <http://www.ode.org/>
- Thornton, S. & Marion, J. (2003). Classical Dynamics of Particles and Systems, Books Cole, 5th Edition
- Trinkle, J.; Pang, J.; Sudarsky, S. & Lo, G. (1997). On Dynamic Multi-Rigid-Body Contact Problems with Coulomb Friction, In: Zeitschrift für Angewandte Mathematik, pages 267 - 279
- Vondrak, M. (2006). Crisis Physics Engine, In: <http://crisis.sourceforge.net/>
- Vondrak, M.; Sigal, L. & Jenkins, C. (2008). Physical Simulation for Probabilistic Motion Tracking, In: Computer Vision and Pattern Recognition (CVPR 2008), pages 1 - 8
- Yin, K.; Coros, S.; Beaudoin, P. & van de Panne, M. (2008). Continuation Methods for Adapting Simulated Skills, In: ACM Transactions on Graphics (SIGGRAPH)
- Yin, K.; Loken, K. & van de Panne, M. (2007). SIMBICON: Simple Biped Locomotion Control, In: ACM Transactions on Graphics (SIGGRAPH)



Motion Control

Edited by Federico Casolo

ISBN 978-953-7619-55-8

Hard cover, 590 pages

Publisher InTech

Published online 01, January, 2010

Published in print edition January, 2010

The book reveals many different aspects of motion control and a wide multiplicity of approaches to the problem as well. Despite the number of examples, however, this volume is not meant to be exhaustive: it intends to offer some original insights for all researchers who will hopefully make their experience available for a forthcoming publication on the subject.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Marek Vondrak, Leonid Sigal and Odest Chadwicke Jenkins (2010). Dynamics and Control of Multibody Systems, Motion Control, Federico Casolo (Ed.), ISBN: 978-953-7619-55-8, InTech, Available from: <http://www.intechopen.com/books/motion-control/dynamics-and-control-of-multibody-systems>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen