Dynamics and Control of Multibody Systems

Marek Vondrak[1], Leonid Sigal[2] and Odest Chadwicke Jenkins[1]
[1]Brown University,
[2]Univesity 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.

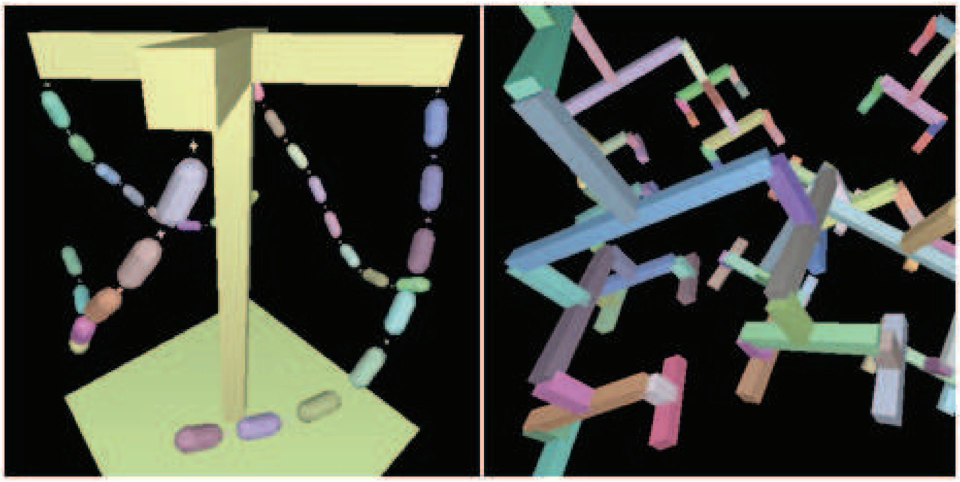

Fig. 1. Examples of constrained rigid body systems. Constraints glue bodies together at designated points, actuate the structures or enforce non-penetration.

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 computationaly 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* $\rho: \boldsymbol{R}^3 \mapsto \boldsymbol{R}^+$ 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* $m = \int \rho(\vec{r}^b)\, \mathrm{d}\vec{r}^b$, *center of mass* $\vec{r}^b_{cm} = \int \frac{\vec{r}^b \rho(\vec{r}^b)}{M}\, \mathrm{d}\vec{r}^b$, *principal moments of inertia* $I_{xx} = \int \left(\left(\vec{r}^b_y\right)^2 + (\vec{r}^b_z)^2\right)\rho(\vec{r}^b)\, \mathrm{d}\vec{r}^b$, $I_{yy} = \int ((\vec{r}^b_x)^2 + (\vec{r}^b_z)^2)\rho(\vec{r}^b)\, \mathrm{d}\vec{r}^b$, $I_{zz} = \int \left((\vec{r}^b_x)^2 + \left(\vec{r}^b_y\right)^2\right)\rho(\vec{r}^b)\, \mathrm{d}\vec{r}^b$ and *products of inertia* $I_{xy} = \int \left(\vec{r}^b_x \vec{r}^b_y\right)\rho(\vec{r}^b)\, \mathrm{d}\vec{r}^b$, $I_{xz} = \int (\vec{r}^b_x \vec{r}^b_z)\rho(\vec{r}^b)\, \mathrm{d}\vec{r}^b$, $I_{yz} = \int \left(\vec{r}^b_y \vec{r}^b_z\right)\rho(\vec{r}^b)\, \mathrm{d}\vec{r}^b$ 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, $\vec{r}^{b}_{cm} = \vec{0}$, 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 $[rad \cdot s^{-1}]$ 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 $\vec{r}$.

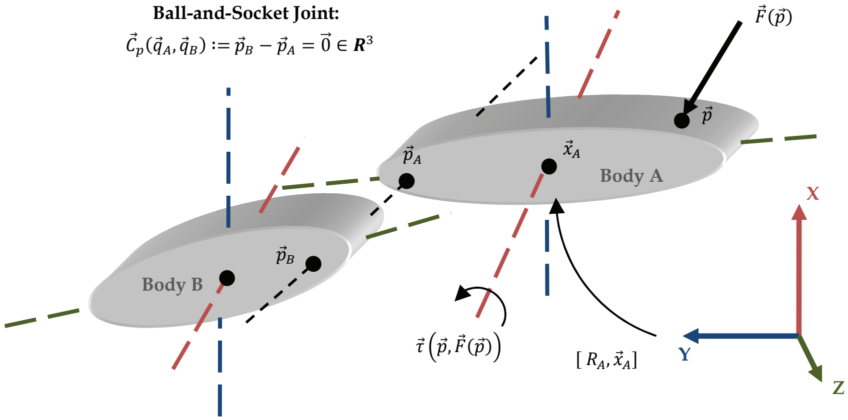

Fig. 2. Illustration of the two constrained bodies in motion.

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* $\left[\vec{F}(\vec{p}),\ \vec{\tau}\left(\vec{p}, \vec{F}(\vec{p})\right)\right]$, where $\vec{\tau}\left(\vec{p}, \vec{F}(\vec{p})\right) = (\vec{p} - \vec{x}) \times \vec{F}(\vec{p})$ 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* $\vec{F}_{\text{total}} = \sum_{\text{i}} \vec{\text{F}}_{\text{i}}$ 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 $\vec{y}_1, \cdots, \vec{y}_n$ into a single state vector $\vec{y} = (\vec{y}_1, \cdots, \vec{y}_n)$ and letting $\frac{\partial \vec{y}}{\partial t} = \left(\frac{\partial \vec{y}_1}{\partial t}, \cdots, \frac{\partial \vec{y}_n}{\partial t}\right)$.

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

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

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 single*[1] *rigid body in the velocity form*

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

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 $\boldsymbol{R}^6$, 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* $\vec{F}_{gen}^{total} = \left(\vec{F}_{total}, \vec{\tau}_{total}\right)$ 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,

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

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 to 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

$$M \cdot \vec{a} = \vec{F}_{total} \tag{4}$$

[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 $\boldsymbol{R}^6$.

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 acceletation. 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 $\vec{C}_p(\vec{q}_1, \vec{q}_2) := \vec{p}_2 - \vec{p}_1 = \vec{0} \in \boldsymbol{R}^3$, 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, $\ddot{\vec{C}}_p = \dot{\vec{C}}_v = \frac{\partial}{\partial t}(-\dot{\vec{x}}_1 - \vec{\omega}_1 \times \vec{r}_1) + \frac{\partial}{\partial t}(\dot{\vec{x}}_2 - \vec{\omega}_2 \times \vec{r}_2) = \left(-\ddot{\vec{x}}_1 - \dot{\vec{\omega}}_1 \times \vec{r}_1 - \vec{\omega}_1 \times (\vec{\omega}_1 \times \vec{r}_1)\right) + \left(\ddot{\vec{x}}_2 + \dot{\vec{\omega}}_2 \times \vec{r}_2 + \vec{\omega}_2 \times (\vec{\omega}_2 \times \vec{r}_2)\right) = \left(-\ddot{\vec{x}}_1 + \vec{r}_1^* \cdot \dot{\vec{\omega}}_1 - \vec{\omega}_1 \times (\vec{\omega}_1 \times \vec{r}_1)\right) + \left(\ddot{\vec{x}}_2 - \vec{r}_2^* \cdot \dot{\vec{\omega}}_2 + \vec{\omega}_2 \times (\vec{\omega}_2 \times \vec{r}_2)\right) = (-E \quad \vec{r}_1^* \quad E \quad -\vec{r}_2^*) \cdot \begin{pmatrix} \vec{a}_1 \\ \vec{a}_2 \end{pmatrix} - \vec{\omega}_1 \times (\vec{\omega}_1 \times \vec{r}_1) + \vec{\omega}_2 \times (\vec{\omega}_2 \times \vec{r}_2)$ 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 $J_1 \cdot \vec{a}_1 + J_2 \cdot \vec{a}_2 = \vec{c} \in \boldsymbol{R}^3$ 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 $(J_1 \quad J_2)$ 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 $\vec{a} = (\vec{a}_1, \vec{a}_2)$, $J = (J_1 \quad J_2)$, $\vec{F}_{total} = (\vec{F}_1^{total}, \vec{F}_2^{total})$, $\vec{F}_c = J^T \cdot \vec{\lambda} = (\vec{F}_1^c, \vec{F}_2^c)$, $M = \begin{pmatrix} M_1 & 0 \\ 0 & M_2 \end{pmatrix}$ 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 $\vec{\lambda} \in \boldsymbol{R}^3$ 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 $\vec{F}_c^i = J_i^T \cdot \vec{\lambda}_i = \left(\left(\vec{F}_c^i\right)_1, \dots, \left(\vec{F}_c^i\right)_n\right)$ 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 $\left(\vec{\lambda}_i\right)_k \cdot (J_i)_k$ on the system. Due to the way J_i is defined, $\left(\vec{F}_c^i\right)_{A_i} = J_{i,A_i}^T \cdot \vec{\lambda}_i$ and $\left(\vec{F}_c^i\right)_{B_i} = J_{i,B_i}^T \cdot \vec{\lambda}_i$ are the only non-zero blocks of $\vec{F}_c^i$ and $\left(\vec{F}_c^i\right)_{A_i}$ is the constraint force applied to the first body and $\left(\vec{F}_c^i\right)_{B_i}$ 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 $A_{i_1,i_2} = \sum_j J_{i_1,j} \cdot M_j^{-1} \cdot \left(J_{i_2,j}\right)^T$. 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 $\vec{C}_p\left(\vec{q}_{A_i}, \vec{q}_{B_i}\right) = \vec{0}$ or a velocity-level constraint $\vec{C}_v\left(\vec{v}_{A_i}, \vec{v}_{B_i}\right) = \vec{0}$ 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 $J_i \cdot \vec{a} - \vec{c}_i = (J \cdot \vec{a} - \vec{c})_i = \left(J \cdot M^{-1} \cdot \vec{F}_{total} + J \cdot M^{-1} \cdot J^T \cdot \vec{\lambda} - \vec{c}\right)_i = \left((J \cdot M^{-1} \cdot J^T) \cdot \vec{\lambda} + (J \cdot M^{-1} \cdot \vec{F}_{total} - \vec{c})\right)_i = \left(A \cdot \vec{\lambda} + \vec{b}\right)_i = A_i \cdot \vec{\lambda} + \vec{b}_i = \vec{0}$. Hence we get that equality constraint i requires that

$$A_i \cdot \vec{\lambda} + \vec{b}_i = \vec{0} \tag{5}$$

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 ≥ or a less-than-or-equal sign ≤. 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, $\left(J_{i,A_i} \cdot \vec{a}_{A_i} + J_{i,B_i} \cdot \vec{a}_{B_i} - \vec{c}_i\right)_k > 0$, then the constraint force due to that hypersurface must vanish, that is $\left(\vec{\lambda}_i\right)_k = 0$, 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,

$$A_i \cdot \vec{\lambda} + \vec{b}_i \geq \vec{0}$$
$$\vec{\lambda}_i \geq \vec{0} \qquad (6)$$
$$\left(A_i \cdot \vec{\lambda} + \vec{b}_i\right) \cdot \vec{\lambda}_i = 0,$$

where $\left(A_i \cdot \vec{\lambda} + \vec{b}_i\right) \cdot \vec{\lambda}_i = \sum_{k=1}^{m_i}\left(A_i \cdot \vec{\lambda} + \vec{b}_i\right)_k \cdot \left(\vec{\lambda}_i\right)_k = 0$ in fact means that $\left(A_i \cdot \vec{\lambda} + \vec{b}_i\right)_k \cdot \left(\vec{\lambda}_i\right)_k$ 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 $\vec{\lambda}_i$,

$$A_i \cdot \vec{\lambda} + \vec{b}_i \leq \vec{0}$$
$$\vec{\lambda}_i \leq \vec{0} \qquad (7)$$
$$\left(A_i \cdot \vec{\lambda} + \vec{b}_i\right) \cdot \vec{\lambda}_i = 0.$$

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 $\| J_i^T \| \cdot \left|\left(\vec{\lambda}_i\right)_1\right|$ 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 $\left(\vec{\lambda}_i\right)_1$. 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 $\left(\vec{\lambda}_i^{lo}\right)_k \leq \left(\vec{\lambda}_i\right)_k \leq \left(\vec{\lambda}_i^{hi}\right)_k$ 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

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

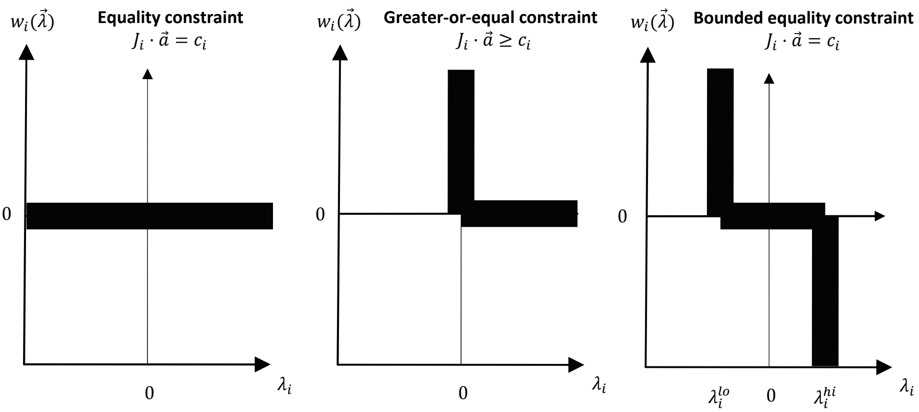

Fig. 3. Visualization of complementarity conditions on the pairs $(\lambda_i, w_i(\vec{\lambda}))$ due to different kinds of one dimensional constraints i, where $w_i(\vec{\lambda}) := A_i \cdot \vec{\lambda} + b_i = J_i \cdot \vec{a} - c_i$. Thick lines indicate permissible values for the $(\lambda_i, w_i(\vec{\lambda}))$ 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, $\overrightarrow{\ }$ 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 $\left(\vec{\lambda}_i\right)_1$, 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 $A \cdot \vec{\lambda} + \vec{b} \geq \vec{0}$, $\vec{\lambda} \geq \vec{0}$, $\vec{\lambda} \cdot \left(A \cdot \vec{\lambda} + \vec{b}\right) = \vec{0}$, 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* $A_{eq} \cdot \vec{\lambda} + \vec{b}_{eq} = \vec{0}, A_{ineq} \cdot \vec{\lambda} + \vec{b}_{ineq} \geq \vec{0}, \vec{\lambda}_{ineq} \geq \vec{0}, \vec{\lambda}_{ineq} \cdot \left(A_{ineq} \cdot \vec{\lambda} + \vec{b}_{ineq}\right) = 0$, 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

$$M \cdot \vec{v} = \vec{F}_{imp} \tag{9}$$

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 $\left(\vec{F}_{imp}^{total}\right)_1, \dots, \left(\vec{F}_{imp}^{total}\right)_n$ 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 $\vec{F}_{imp}^{total} = \left(\left(\vec{F}_{imp}^{total}\right)_1, \dots, \left(\vec{F}_{imp}^{total}\right)_n\right)$. 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 $\left(\vec{F}_c^i\right)_{imp} = J_i^T \cdot \vec{\lambda}_i$ 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 $\left(\vec{\lambda}_i^{lo}\right)_k \leq \left(\vec{\lambda}_i\right)_k \leq \left(\vec{\lambda}_i^{hi}\right)_k$ 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 $\vec{k} = \left(\vec{k}_1, \ldots, \vec{k}_c\right)$.

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 $\vec{C}_p^i\left(\vec{q}_{A_i}, \vec{q}_{B_i}\right) \in \boldsymbol{R}^{m_i}$ 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 $\vec{C}_p^i\left(\vec{q}_{A_i}, \vec{q}_{B_i}\right) = \vec{0}$ for generalized positions $\vec{q}_{A_i}$ and $\vec{q}_{B_i}$ and the value of $\vec{C}_p^i\left(\vec{q}_{A_i}, \vec{q}_{B_i}\right)$ 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 $\vec{C}_p^i\left(\vec{q}_{A_i}, \vec{q}_{B_i}\right) \geq \vec{0}$ and *position level less-or-equal constraint* i requests that $\vec{C}_p^i\left(\vec{q}_{A_i}, \vec{q}_{B_i}\right) \leq \vec{0}$.

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 $\vec{C}_p^i = \vec{0}$. 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

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

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

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

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