

Ideal constraints in real-time simulation

Miguel González Cuadrado

2nd April 2018

Abstract

A method is described for dealing with ideal constraints in real-time simulation. The method requires that the system simulated be described by a set of ordinary differential equations, augmented by a set of constraint conditions in the form of implicit equations on the state vector. The constraint terms (constraint forces and moments in the case of dynamics) are computed and can be used by the rest of the physical model.

Contents

1	The problem	1
2	Virtual displacements and ideal constraints	3
3	Justification of the ideal constraint condition	4
4	Determination of the Lagrange multiplier	6
5	Constraint tracking	6
6	Putting it all together	7
7	Constraint terms	7

1 The problem

In real-time simulation, like in many other simulation tasks, the evolution of the system considered is computed through the numerical integration of the

differential equations that rule the system. In most cases, the differential equations are ordinary differential equations (ODE), and can be put in the following form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \mathbf{u}; t) \quad (1.1)$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is an input vector, and t is time. We will call n the dimension of $\mathbf{x}$. The vector function $\mathbf{f}$ has dimension n too, whereas $\mathbf{u}$ has any dimension.

When constraints act upon the system, they can be modeled as

$$\mathbf{g}(\mathbf{x}; \mathbf{u}; t) = \mathbf{0} \quad (1.2)$$

where the dimension of $\mathbf{g}$ is m , which is also the number of one-dimensional constraints (i.e. the number of degrees of freedom lost by the system as a consequence of the constraints).

When the simulated system is for example a rigid body, the constraints can be in terms of positions, attitude, velocity, angular velocity, or any combination thereof.

The constraints introduce constraint terms in equation (1.1), in the form of additional contributions to $\dot{\mathbf{x}}$:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \mathbf{u}; t) + \mathbf{h}(\mathbf{x}; \mathbf{u}; t) \quad (1.3)$$

In the example of the rigid body, $\mathbf{h}$ is analogous to forces and moments when the constraint is on velocity and angular velocity; when the constraint is on position and attitude, $\mathbf{h}$ consists of additional velocity and angular velocity in the integration of position and attitude¹.

With the introduction of constraints and constraint terms, there appears an indetermination due to the fact that the constraint terms are not known. This is solved by adding additional conditions that limit the dimension of $\mathbf{h}$. Usually, for dynamics problems the conditions state that the constraints are ideal. Ideal constraints are those constraints whose constraint actions develop no work *for any virtual displacement compatible with the*

¹Strictly speaking, these additional velocity and angular velocity have no physical meaning, whereas constraint forces and moments are real forces and moments. Nevertheless, they may be interpreted as force and moment impulses, e.g. when computing non-ideal actions, which can be modeled as ideal actions plus additional non-ideal terms that depend on the magnitude of the ideal actions.

constraints. Mathematically, a virtual displacement $\delta \mathbf{x}$ is compatible with the constraints $\mathbf{g}(\mathbf{x}; \mathbf{u}; t)$ if it belongs into the kernel of $\delta \mathbf{g} = \mathbf{0}$.

We will momentarily assume that a similar definition applies to the constraint term of equation (1.3). Namely, ideal constraints are defined as those constraints whose constraint terms develop no work for any virtual displacement compatible with the constraints, with the “work” of $\mathbf{h}$ for a virtual displacement $\delta \mathbf{x}$ being $\mathbf{h}(\mathbf{x}; \mathbf{u}; t) \cdot \delta \mathbf{x}$, this is, in matrix form, $\mathbf{h}(\mathbf{x}; \mathbf{u}; t)^T \delta \mathbf{x}$. The virtual displacements compatible with the constraints are defined as before. This definition for ideal constraints will be justified in section 3.

2 Virtual displacements and ideal constraints

In order to determine which are the virtual displacements compatible with the constraints, which we will call VDCCs hereafter, we must develop $\delta \mathbf{g}$:

$$\delta \mathbf{g} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial \mathbf{g}}{\partial \mathbf{u}} \delta \mathbf{u} + \frac{\partial \mathbf{g}}{\partial t} \delta t \quad (2.1)$$

where $\frac{\partial \mathbf{a}}{\partial \mathbf{b}}$ is the jacobian of $\mathbf{a}$ with respect to $\mathbf{b}$, and the virtual displacement $\delta \mathbf{x}$ corresponds to virtual input and time increments $\delta \mathbf{u}$ and δt respectively.

For brevity, let

$$\mathbf{g}_x = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \quad (2.2)$$

and let us drop the list of arguments off the functions.

A virtual displacement $\delta \mathbf{x}$ that satisfies $\delta \mathbf{g} = \mathbf{0}$ is said to be a *possible displacement*. The kernel of $\delta \mathbf{g} = \mathbf{0}$ is the set spanned by the differences between any two possible displacements. If $\delta \mathbf{x}_1$ and $\delta \mathbf{x}_2$ are possible displacements, then

$$\mathbf{0} = \delta \mathbf{g}_1 - \delta \mathbf{g}_2 = \mathbf{g}_x (\delta \mathbf{x}_1 - \delta \mathbf{x}_2) \quad (2.3)$$

Therefore, since the possible displacement span affine hyperplanes, a virtual displacement $\delta \mathbf{x}$ is a VDCC if and only if

$$\mathbf{g}_x \delta \mathbf{x} = \mathbf{0} \quad (2.4)$$

The condition of the constraints being ideal can then be stated as

$$\forall \delta \mathbf{x}, (\mathbf{g}_x \delta \mathbf{x} = \mathbf{0} \implies \mathbf{h}(\mathbf{x}; \mathbf{u}; t)^T \delta \mathbf{x} = \mathbf{0}) \quad (2.5)$$

which implies that the column-vector $\mathbf{h}(\mathbf{x}; \mathbf{u}; t)$ is a linear combination of the rows of $\mathbf{g}_x$, that is

$$\mathbf{h}(\mathbf{x}; \mathbf{u}; t) = \mathbf{g}_x^T \boldsymbol{\lambda} \quad \text{for some value of the vector } \boldsymbol{\lambda} \quad (2.6)$$

The vector $\boldsymbol{\lambda}$ is called “Lagrange multiplier”.

This equation can be introduced into the differential equation for the system with the additional constraint terms (1.3), to obtain

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \mathbf{u}; t) + \mathbf{g}_x^T \boldsymbol{\lambda} \quad (2.7)$$

This introduces n new unknowns into the differential equation for the system (1.1), that cancel out the n new constraints introduced by the constraint equation (1.2).

3 Justification of the ideal constraint condition

In order for the definition of ideal constraint to be useful, we must ensure that it is equivalent to the definition of ideal constraint in dynamics.

In dynamics, the state of the system is described by a set of generalised coordinates $\mathbf{r}$ and their derivatives $\mathbf{v} = \dot{\mathbf{r}}$. The differential equations that rule the system are the following second-order ODE:

$$\ddot{\mathbf{r}} = \mathbf{q} \quad (3.1)$$

where $\mathbf{q}$ are the generalised forces. This second-order ODE can be converted into a first-order ODE in terms of a single state vector $\mathbf{x}$ composed of $\mathbf{r}$ and $\mathbf{v}$:

$$\mathbf{x} = \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \end{pmatrix} \quad (3.2)$$

$$\dot{\mathbf{x}} = \begin{pmatrix} 0 & I \\ 0 & 0 \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{0} \\ \mathbf{q} \end{pmatrix} \quad (3.3)$$

The constraints introduce constraint forces $\mathbf{s}$:

$$\dot{\mathbf{x}} = \begin{pmatrix} 0 & I \\ 0 & 0 \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{0} \\ \mathbf{q} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{s} \end{pmatrix} \quad (3.4)$$

We must now make the difference between *finite* and *kinematic* constraints. Finite constraints are those that do not depend on $\mathbf{v}$, i.e. those for

which $\frac{\partial g_i}{\partial \mathbf{v}} = \mathbf{0}$. Kinematic constraints are non-finite constraints. Let us separate finite constraints from kinematic constraints:

$$\mathbf{g}(\mathbf{r}, \mathbf{v}; \mathbf{u}; t) = \begin{pmatrix} \mathbf{g}^f(\mathbf{r}; \mathbf{u}; t) \\ \mathbf{g}^k(\mathbf{r}, \mathbf{v}; \mathbf{u}; t) \end{pmatrix} \quad (3.5)$$

In dynamics, it is shown, by a reasoning similar to what is explained in section 2, that the constraint forces $\mathbf{s}$ have the following form:

$$\mathbf{s} = \begin{pmatrix} \mathbf{g}_r^f \\ \mathbf{g}_v^k \end{pmatrix} \lambda \quad (3.6)$$

In the first-order ODE case, finite constraints will yield constraint terms that act upon $\mathbf{r}$, whereas constraint terms corresponding to kinematic constraints act upon $\mathbf{r}$ and $\mathbf{v}$. If finite constraint terms are to be analogous to forces, they must be derived so that their constraint terms act upon $\mathbf{v}$:

$$\mathbf{g}^f(\mathbf{r}; \mathbf{u}; t) = \mathbf{0} \quad \Rightarrow \quad \mathbf{0} = \frac{d\mathbf{g}^f}{dt} = \left(\mathbf{g}_r^f\right)^T \mathbf{v} + \left(\mathbf{g}_u^f\right)^T \dot{\mathbf{u}} + \mathbf{g}_t^f \quad (3.7)$$

The finite constraint $\mathbf{g}^f$ is replaced by its derivative with respect to time, whose jacobian with respect to $\mathbf{v}$ is $\mathbf{g}_r^f$. This leads to a constraint term of the form

$$\mathbf{h}(\mathbf{x}; \mathbf{u}; t) = \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} \lambda \quad (3.8)$$

where $\mathbf{s}_1$ acts upon $\mathbf{r}$, and $\mathbf{s}_2$ acts upon $\mathbf{v}$, and

$$\mathbf{s}_2 = \begin{pmatrix} \mathbf{g}_r^f \\ \mathbf{g}_v^k \end{pmatrix} \quad (3.9)$$

Therefore, if finite constraints are replaced by their derivatives with respect to time, constraint terms corresponding to $\mathbf{v}$ are the same as the generalised forces that arise from them in the dynamics treatment.

On the other hand, the constraint terms $\mathbf{s}_1$ that act upon $\mathbf{r}$ have no counterpart in the dynamics treatment, because they correspond to instantaneous increments in the integration that yields $\mathbf{r}$, i.e. they are analogous to increments in velocity that are not the result of forces, constraint or otherwise. There are two ways to deal with this discrepancy. First, we can pretend in the rest of this explanation, except for constraint tracking (section 5), that $\mathbf{g}_r \equiv \mathbf{0}$, and forget about the underived finite constraints; this

will yield results equivalent to dynamics theory. Second, we can use the additional stabilisation provided by the terms on $\mathbf{r}$, and, instead of replacing the finite constraints, augment them with their derivatives with respect to time; this will not be equivalent to dynamics theory, but will yield a more stable numerical integration.

4 Determination of the Lagrange multiplier

The Lagrange multiplier $\boldsymbol{\lambda}$ in equation (2.7) can be determined by deriving the constraint equation (1.2) with respect to time:

$$\mathbf{0} = \frac{d\mathbf{g}}{dt} = \mathbf{g}_x \dot{\mathbf{x}} + \mathbf{g}_u \dot{\mathbf{u}} + \mathbf{g}_t \quad (4.1)$$

which, introduced into equation (2.7) yields

$$\mathbf{g}_x \dot{\mathbf{x}} = \mathbf{g}_x \mathbf{f}(\mathbf{x}; \mathbf{u}; t) + \mathbf{g}_x \mathbf{g}_x^T \boldsymbol{\lambda} = -\mathbf{g}_u \dot{\mathbf{u}} - \mathbf{g}_t \quad (4.2)$$

hence,

$$\mathbf{g}_x \mathbf{g}_x^T \boldsymbol{\lambda} = -\mathbf{g}_x \mathbf{f}(\mathbf{x}; \mathbf{u}; t) - \mathbf{g}_u \dot{\mathbf{u}} - \mathbf{g}_t \quad (4.3)$$

Although $\mathbf{g}_x$ is not generally square, $\mathbf{g}_x \mathbf{g}_x^T$ is square, of dimension $m \times m$. If $\mathbf{g}_x \mathbf{g}_x^T$ is invertible, the value of $\boldsymbol{\lambda}$ can be obtained from (4.3):

$$\boldsymbol{\lambda} = -(\mathbf{g}_x \mathbf{g}_x^T)^{-1} (\mathbf{g}_x \mathbf{f} + \mathbf{g}_u \dot{\mathbf{u}} + \mathbf{g}_t) \quad (4.4)$$

using which we can compute the constraint term $\mathbf{h}$ per equation (2.6), and introduce it in equation (1.1).

5 Constraint tracking

Even though the described scheme maintains the constraints and ensures the constraint equation (1.2) in the mathematical sense, in practice numerical integration and limited accuracy will make the system drift off the constraint condition. In order to avoid this, the system state can be forced *after each numerical integration step* to fulfill the constraints. This can be implemented in the form of a finite correction to the state vector, with the condition that the finite correction be analogous to an ideal constraint term:

$$\mathbf{x} \leftarrow \mathbf{x}^+ = \mathbf{x} + \Delta \mathbf{x} \quad (5.1)$$

with

$$\Delta \mathbf{x} = \mathbf{g}_x^T \boldsymbol{\mu} \quad \text{for some value of the vector } \boldsymbol{\mu} \quad (5.2)$$

The condition on $\boldsymbol{\mu}$ is that

$$\mathbf{g}(\mathbf{x}^+; \mathbf{u}; t) = \mathbf{0} \quad (5.3)$$

which, linearising around $\mathbf{x}$ yields

$$\mathbf{g}(\mathbf{x}; \mathbf{u}; t) + \mathbf{g}_x \Delta \mathbf{x} = \mathbf{0} \quad (5.4)$$

hence

$$\boldsymbol{\mu} = -(\mathbf{g}_x \mathbf{g}_x^T)^{-1} \mathbf{g}(\mathbf{x}; \mathbf{u}; t) \quad (5.5)$$

6 Putting it all together

This numerical scheme for integration of differential equations with constraints turns

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \mathbf{u}; t) \quad (6.1)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}; \mathbf{u}; t) \quad (6.2)$$

using the hypothesis of ideal constraint term, into

$$H = (\mathbf{g}_x \mathbf{g}_x^T)^{-1} \quad (6.3)$$

$$\mathbf{h} = -\mathbf{g}_x^T H (\mathbf{g}_x \mathbf{f}(\mathbf{x}; \mathbf{u}; t) + \mathbf{g}_u \dot{\mathbf{u}} + \mathbf{g}_t) \quad (6.4)$$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \mathbf{u}; t) + \mathbf{h} \quad (6.5)$$

$$\mathbf{x} \leftarrow \mathbf{x} - \mathbf{g}_x H \mathbf{g}(\mathbf{x}; \mathbf{u}; t) \quad (6.6)$$

where the last operation (6.6) is to be performed after each numerical integration step.

7 Constraint terms

When the constraint terms are to be used in the rest of the physical model (e.g. for computing friction forces), they are available as $\mathbf{h}$ in equation (6.4).