Energy-stable Modelling of Contacting Modal Objects With Piecewise-Linear Interaction Force

Matthias Rath

Berlin University of Technology
Deutsche Telekom Laboratories, Quality and Usability

September 2, 2008 / DAFx08, Helsinki

Overview

Motivation and scope

Background, situation, terms

- Modelling impact interaction using force laws
- Facts about differential equations with linear operators
- Systems with finite-dimensional state space

Modelling contact by modal description

- Principle idea
- Contact of finite-dimensional systems
- Discrete-time realization of contact condition
- Example implementation

Conclusions and outlook

Context and motivation

Interactive realtime synthesis of sounds of contacting solid objects: dynamic modelling of *continuous* and interrupted contact? (In particular: rolling. . .)

- Modal description successful for many cases of *one single* solid object — under clear conditions of mathematical theory. . .
- Used approaches for force signals: impulsive force approximation for very short contact, ad-hoc assumptions for longer single contacts (e.g. cosine half-periods. . .), force profiles for continuous interaction on the basis of more or less heuristic ideas. . .
- *But*: situations of *varying, continuous and interrupted* contact?
  - E.g.: ball falling, bouncing, rolling
  - Possible answer: models *including interaction forces*. . .

Motivation, specific scope

Modelling of rolling sound through contact perpendicular to plane of contact. . . :

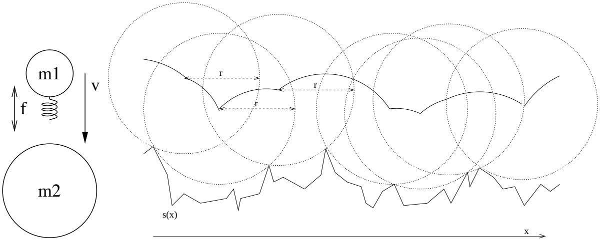


Figure: Reduction of rolling geometry to 1-dimensional impact.

Summary: necessity of a computational model of impact accounting for "microscopic" interaction behavior *and* macroscopic/long-term energy preservation.

Background – Modelling impact interaction using force laws *force laws*

Different laws have been proposed for the interrelation of "global distances" and the interaction force acting on the involved objects, e.g. by Hertz or by Hunt and Crossley:

$$f(x, \dot{x}) = \begin{cases} -kx^{\alpha} - \lambda x^{\alpha} \cdot \dot{x}, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{1}$$

- Non-linearity of scenario, therefore: discrete-time implementation by *numerical approximation*
- *But*: even small inexactnesses lead to uncontrolled changes of the energy stored in the system and may add up in long-term behavior.
  - (The system *as a whole* is necessarily non-linear, *even when linearizing* the first line of equation (1).)

Central ideas: *piece-wise linear* behaviour of interaction force, *exact* solution in each linear stage, *appropriate "switching"*.

Background – general principles

Description of the temporal behaviour of a system

$$\dot{\vec{z}}(t) = A\vec{z}(t) + \vec{f}_{\text{ext}}(t), \tag{2}$$

$\vec{z}(t)$ state of the system in the "*state space*" vector space Z — thus *state "vector"*, A a *linear* operator defined on state space.

Remarks:

- *The state space may be finite- or also infinite-dimensional*.
- The demand on A to be linear is central for the term of "modes of a system" to make sense.
- Core idea of modal approach: simplify and solve equation (2) by expressing the state vector $\vec{z}$ in a basis of (generalised) eigenvectors of A.
  - E.g.: if $\vec{z}$ is an eigenvector of A to eigenvalue d, (2) reduces to $\dot{\vec{z}}(t) = d \cdot \vec{z}(t)$ *with d scalar!* (no external forces, $\vec{f}_{\text{ext}}(t) = \vec{0}$, homogeneous version).

Background – Finite-dimensional state space

Finite-dimensional state space: operator A "is" matrix. General solution (homogeneous case)

$$\vec{z}(t) = e^{tA}\vec{z}(0) \tag{3}$$

then allows for an *exact* (up to resolution of finite computer architecture) computation in discrete-time algorithm by means of the "*(state) transition matrix*" $e^{t_\Delta A}$ (to the time step t_Δ).

- In this case the *modal approach* of finding the generalised eigenvectors of A consists in finding a similarity transformation
$$N := V^{-1}AV \tag{4}$$
to Jordan canonical form, V non-singular.
  - In most practical cases Jordan canonical form is diagonal, i.e. $A = VDV^{-1}(\Leftrightarrow D = V^{-1}AV)$, D diagonal matrix. Transition matrix: $e^{t_\Delta A} = Ve^{t_\Delta D}V^{-1}$, with $e^{t_\Delta D}$ also diagonal.

Background – Stiffness and friction matrices

Most common practical application: system of second order differential equations of the form

$$M\ddot{\vec{x}}(t) + K\dot{\vec{x}}(t) + C\vec{x}(t) = \vec{f}^{\text{ext}}(t), \tag{5}$$

M, K and C matrices representing inertia, “stiffness” and “friction”.

- Reduction: $\vec{z} := \begin{pmatrix} \vec{x} \\ \dot{\vec{x}} \end{pmatrix}$, $A := \begin{pmatrix} O & E \\ -K & -C \end{pmatrix}$
- Most often derived from Newton’s laws for an idealised system of lumped masses or from Lagrange equations, but may also be derived from a spatially continuous system which has first been transformed directly by the modal approach and then simplified.
  - Inertia matrix M is generally invertible (most often diagonal) and may therefore be omitted (K and C to $M^{-1}K$ and $M^{-1}C$).

Modelling contact by modal description

Two systems of the described type of equation (2)

$$\dot{\vec{z}}_1(t) = A_1\vec{z}_1(t) + \vec{f}_{1\text{ext}} \quad \text{and} \quad \dot{\vec{z}}_2(t) = A_2\vec{z}_2(t) + \vec{f}_{2\text{ext}} \qquad (6)$$

interacting by means of some interaction force $\vec{f}(\vec{z}_1, \vec{z}_2)$.
For simplicity:

- no other external forces,
- "actio–reactio" principle holds: $\vec{f}_1 (= \vec{f}_{1\text{ext}}) = \vec{f}$ and $\vec{f}_2 = -\vec{f}$.

General idea: If $\vec{f}$ is a *linear* function of $\vec{z}_1$ and $\vec{z}_2$, the system of both masses *during contact* may again be written in the form $\dot{\vec{z}}(t) = A\vec{z}(t)$ with a *linear* operator A on the state space $Z = Z_1 \times Z_2$, "built from" A_1, A_2 and the linear expression of $\vec{f}$.

Contact of finite-dimensional systems – example, point–mass “hammer”

Modal object as in equation (5) struck by a point-mass “hammer” (to keep the overall system possibly simple and demonstrate the general idea),

$$m_h \ddot{x}_{n+1} = f. \tag{7}$$

One contact point l_{con}, contact modelled as a massless damped-spring connection:

$$f = \begin{cases} k_{\text{con}}(x_{l_{\text{con}}} - x_{n+1}) + c_{\text{con}}(\dot{x}_{l_{\text{con}}} - \dot{x}_{n+1}), & \\ \quad\quad\quad\quad\quad\quad\quad\quad\quad \text{for } x_{n+1} < x_{l_{\text{con}}} & \\ 0, \quad\quad\quad\quad\quad\quad\quad\quad \text{otherwise.} & \end{cases} \tag{8}$$

Contact of finite-dimensional systems – example, point–mass “hammer”

Accelerations on hammer and contact point ($f_{l_{\text{con}}} = -f$ according to the actio–reactio principle):

$$\ddot{x}_{n+1} = \frac{k_{\text{con}}}{m_h} x_{l_{\text{con}}} - \frac{k_{\text{con}}}{m_h} x_{n+1} + \frac{c_{\text{con}}}{m_h} \dot{x}_{l_{\text{con}}} - \frac{c_{\text{con}}}{m_h} \dot{x}_{n+1} \tag{9}$$

and

$$\ddot{x}_{l_{\text{con}}} = \frac{k_{\text{con}}}{m_{l_{\text{con}}}} x_{n+1} - \frac{k_{\text{con}}}{m_{l_{\text{con}}}} x_{l_{\text{con}}} + \frac{c_{\text{con}}}{m_{l_{\text{con}}}} \dot{x}_{n+1} - \frac{c_{\text{con}}}{m_{l_{\text{con}}}} \dot{x}_{l_{\text{con}}} + f_{l_{\text{con}}}^{(\text{int})}, \tag{10}$$

$f_{l_{\text{con}}}^{(\text{int})}$ internal force acting on $m_{l_{\text{con}}}$ inside the object.
Combine positions of the n masses of the object with position of the hammer x_{n+1} into $\acute{\vec{x}} := (x_1 \ldots x_{n+1})^{\mathsf{T}}$:

$$\ddot{\acute{\vec{x}}}(t) = -K_{\text{con}} \acute{\vec{x}}(t) - C_{\text{con}} \dot{\acute{\vec{x}}}(t) + \vec{f}^{(\text{int})}(t), \tag{11}$$

with

Contact of finite-dimensional systems – example, point–mass “hammer”

$$K_{\text{con}} := \begin{pmatrix} 0\dots & & & & 0 \\ \vdots\ddots & & & & \\ & \vdots & \frac{k_{\text{con}}}{m} & 0\dots0 & -\frac{k_{\text{con}}}{m} \\ & & 0 & 0\dots0 & 0 \\ & & \vdots & \ddots & \vdots \\ & & 0 & 0\dots0 & 0 \\ 0\dots0 & & -\frac{k_{\text{con}}}{m_h} & 0\dots0 & \frac{k_{\text{con}}}{m_h} \end{pmatrix} \leftarrow l_{\text{con}}\text{th row} \tag{12}$$

C_{con} analogous.
Combine matrices K and C with K_{con} and C_{con} into
$\acute{K} := \begin{pmatrix} K & 0 \\ 0 & 0 \end{pmatrix} + K_{\text{con}}$ and $\acute{C} := \begin{pmatrix} C & 0 \\ 0 & 0 \end{pmatrix} + C_{\text{con}}$, to gain the **final equation of the entire system during contact** as

$$\ddot{\acute{\vec{x}}}(t) + \acute{K}\acute{\vec{x}}(t) + \acute{C}\dot{\acute{\vec{x}}}(t) = \vec{0}. \tag{13}$$

Modelling contact by modal description

Remarks:

- Presented procedure applicable in an analogous way
  - for modal objects given by abstract attributes of modal frequencies, decay times, and weights at the point of contact,
  - if the “hammer” is not as simple as a point mass, but describable by means of a finite-dimensional linear operator.
- The principal idea also holds for systems with finite-dimensional state spaces, but its application is then generally much more involved and no general “receipt” for concrete solution can be given.

Discrete-time realization

Main points:

- Update steps in each linear phase (with transition matrices) are *exact* because based on analytical solution; in particular energy-preserving.
- **Crucial point**: "phase switching" — Exact moments of transition between different linear phases will not be at sample steps.

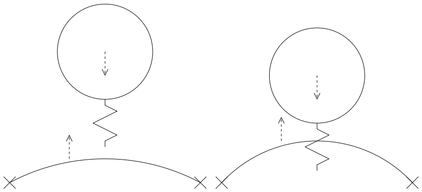


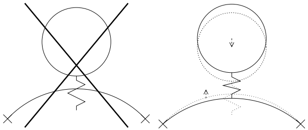
Figure: Single objects at discrete sampled moments just before and after occurence of contact.

How to switch between non-contact and contact?

Discrete-time realization

Unbreakable constraints (here):

- Overlap of objects *must not occur*.
- Inserting a “contact spring” in streched state means erroneous insertion of energy into system.



Present solution: usage of appropriate ideal, massless, perfectly stiff, connecting element.

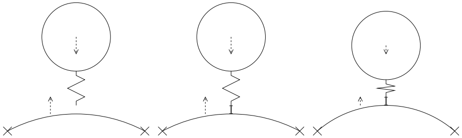


Figure: Energy-preserving update at start of contact

Discrete-time realization

Steps of discrete-time algorithm:

- No contact phase: update seperate objects with individual transition matrices
  - Check if contact starts. If yes: *go back one step* and set contact. Use "*offset element*" to preserve energy.
- Contact phase: update with transition matrix of system in contact.
  - Check if still in contact. If no: release contact. Loss of energy at cut of compressed spring, *but never insertion of energy*.

Example implementation: string struck by “hammer”

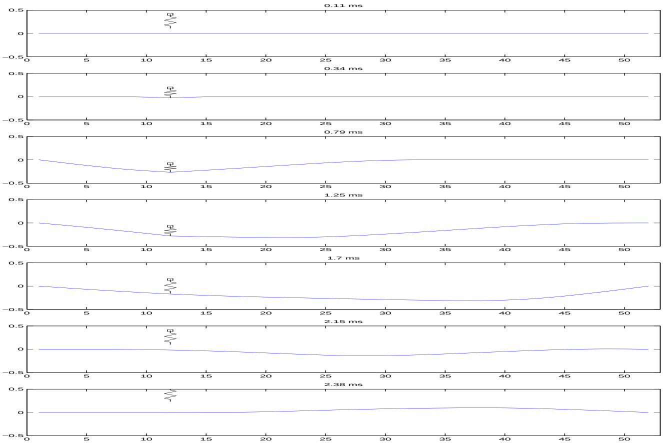
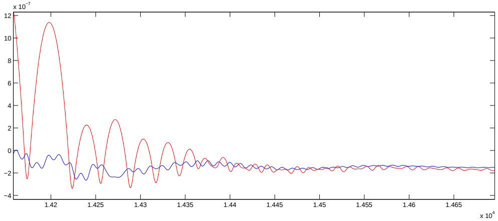


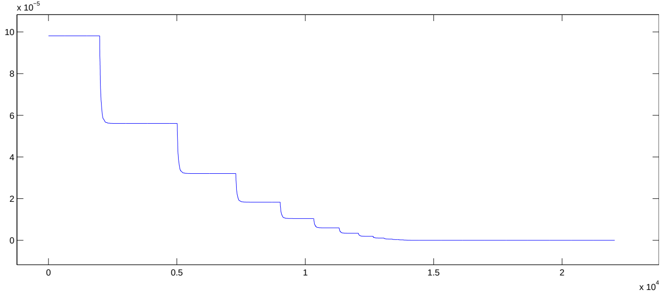
Figure: Snapshots of a string struck by a free mass “hammer”.

Example implementation: string struck by “hammer” – slow motion view

Example implementation: string struck by “hammer” – realtime view, gravity

Example implementation: string struck by “hammer” – continuous contact and energy behaviour





Conclusions and outlook

- An approach and technique has been developed for dynamic modelling of contact interaction, *under complete control of system energy*.
- The model has been implemented and runs at low computational load on standard hardware.
- The current realization is very simple but will be refined for more complex scenarios:
  - Contact force laws with several linear phases approximating laws s.a. Hertz's or Hunt and Crossley's, or stick–slip friction.
  - More complex scenarios than one point-mass.
- Currently offline diagonalization, therefore fixed contact points. Idea: direct online computation of transition matrix...

Thank you!