Recent developments in simulation, optimization and control of flexible multibody systems

Olivier Brüls

Department of Aerospace and Mechanical Engineering
University of Liège
o.bruls@ulg.ac.be

Outline

- Introduction
- Modelling of multibody & mechatronic systems
- Time integration algorithms
- Topology optimization of structural components
Introduction

Some models developed using Samcef/Mecano

Introduction: Typical simulation library

Rigid & flexible joints

Rigid & flexible bodies

Special elements

Force elements
Introduction: Commercial simulation tools

Multibody dynamics approach
- MSC ADAMS
- LMS VIRTUAL LAB MOTION
- SIMPACK
- RECURDYN

Finite element approach
- SAMCEF MECANO
- OOFELIE

- Linear flexibility effects can be represented
- Most general approach for nonlinear flexible systems

Introduction: Mechatronic design

Recent developments in multibody dynamics:
- extension to mechatronic systems
- development of integrated optimization tools
Outline

- Introduction
- Modelling of multibody & mechatronic systems
  - Modelling of flexible multibody systems
  - Modelling of coupled mechatronic systems
  - Application to a semi-active car suspension
- Time integration algorithms
- Topology optimization of structural components

Modelling of flexible multibody systems

Finite element approach [Géradin & Cardona 2001]

Nodal coordinates
  - translations & rotations
  - geometric nonlinearities

Kinematic joints
  - algebraic constraints

\[
M(q)\ddot{q} + g_{gyr}(q, \dot{q}) + g_{int}(q) + \Phi_q^T \lambda = g_{ext} \\
\Phi(q, \dot{q}) = 0
\]

index-3 DAE with rotation variables
Modelling of coupled mechatronic systems

Modular and monolithic FE approach

Flexible Mechanism (FE formalism)

Control System (Block diagram language)

Strongly coupled Eqns

Time integration

Block diagram language in a FE code

- Generic blocks: gain, integrator, transfer function…
  ⇒ “special” elements
- Control state/output variables ⇒ “special” dofs
- Numerical assembly according to the FE procedure
**Modelling of coupled mechatronic systems**

Coupled equations:
\[
\begin{align*}
M(q)\ddot{q} & = g(q, \dot{q}, t) - \Phi_q^T \lambda + Ly \\
0 & = \Phi(q, t) \\
\dot{x} & = f(q, \dot{q}, \dot{q}, \lambda, x, y, t) \\
y & = h(q, \dot{q}, \dot{q}, \lambda, x, y, t)
\end{align*}
\]

Time-integration scheme for coupled 1st/2nd order DAE?
- Classical ODE solvers: multistep & Runge-Kutta methods
- Generalized-α time integration scheme

**Semi-active car suspension**

Work in collaboration with KULeuven-PMA and UCL-CEREM (PA15/6)
Semi-active car suspension

Electrical current in the valves (A)
(rear-left wheel)

Hydraulic pressures (Pa)
(rear-left wheel)
Summary

Strongly coupled simulation of mechatronic systems:

- Mechanical equations are obtained using the finite element technique (rigid bodies, elastic bodies & kinematic joints)

- Control equations are formulated in the FE code using the block diagram language

- The generalized-\( \alpha \) time integrator is used to solve the strongly coupled problem

Outline

- Introduction

- Modelling of multibody & mechatronic systems

  - Time integration algorithms
    - Generalized-\( \alpha \) method
    - Kinematic constraints
    - Treatment of rotation variables
    - Controller dynamics

- Topology optimization of structural components
Numerical time-integration methods
- Standard integrators: multistep, Runge-Kutta
- Methods from structural dynamics (Newmark, HHT, g-α)
- Energy conserving schemes

Generalized-α method [Chung & Hulbert 1993]
- One step method for 2nd ODEs
- 2nd order accuracy
- Unconditional stability (A-stability) for linear problems
- Controllable numerical damping at high frequencies
- Computational efficiency for large and stiff problems

⇒ Extensions of the g-α method to deal with kinematic constraints, rotation variables and controller dynamics?

2nd order ODE system:
\[ M(q)\ddot{q} = g(q, \dot{q}, t) \]

Newmark implicit formulae:
\[ q_{n+1} = q_n + h\ddot{q}_n + \frac{h^2}{2}(0.5 - \beta)\dddot{q}_n + \frac{h^2}{2}\beta a_{n+1} \]
\[ \ddot{q}_{n+1} = \ddot{q}_n + h(1 - \gamma)\dddot{q}_n + h\gamma a_{n+1} \]

Generalized-α method [Chung & Hulbert, 1993]
\[ (1 - \alpha_m) a_{n+1} + \alpha_m a_n = (1 - \alpha_f)\ddot{q}_{n+1} + \alpha_f \dddot{q}_n \]

To be solved with:
\[ M(q_{n+1})\ddot{q}_{n+1} = g(q_{n+1}, \dot{q}_{n+1}, t_{n+1}) \]

- Two kinds of acceleration variables: \( a_n \neq \dddot{q}_n \)
- Algorithmic parameters: \( \gamma, \beta, \alpha_f, \alpha_m \)
  2nd order accuracy & numerical damping
Kinematic constraints

\[
M(q) \ddot{q} - g(q, \dot{q}, t) - \Phi_q^T \lambda = 0 = \Phi(q, t)
\]

Direct integration of the index-3 DAE problem using g-\(\alpha\)

- Linear stability analysis demonstrates the importance of numerical damping [Cardona & Géradin 1989]
- Scaling of equations and variables reduces the sensitivity to numerical errors [Bottasso, Bauchau & Cardona 2007]
- Global convergence is demonstrated [Arnold & B. 2007]

Reduced index formulations

[Lunk & Simeon 2006; Jay & Negrut 2007; Arnold 2009]

Treatment of rotations

It is impossible to have a global 3-dimensional parameterization of rotations without singular points. [Stuelpnagel 1964]

Possible strategies

- 3-dimensional parameterization + reparameterization to avoid singularities [Cardona & Géradin 1989]
- Higher dimensional parameterization + kinematic constraints [Betsch & Steinmann 2001]
- Rotationless formulation, e.g. ANCF [Shabana]
- Lie group time integrator: no parameterization of the manifold is required a priori [Simo & Vu-Quoc 1988; B. & Cardona 2010]
The configuration evolves on the \( n \)-dimensional Lie group
\[
G = \mathbb{R}^3 \times \ldots \times \mathbb{R}^3 \times SO(3) \times \ldots \times SO(3)
\]
with the composition \( q_{\text{tot}} = q_1 \circ q_2 \) such that
\[
x_{i,\text{tot}} = x_{i,1} + x_{i,2} \quad \text{and} \quad R_{i,\text{tot}} = R_{i,1}R_{i,2}
\]

**Constrained equations of motion**:
\[
\begin{align*}
\dot{q} &= DL_q(e) \cdot \dot{v} \\
M(q)\ddot{v} + g(q, v, t) + B^T(q)\lambda &= 0 \\
\Phi(q) &= 0
\end{align*}
\]

**Treatment of rotations**

**Nodal coordinates**
\[
q = (x_1, \ldots, x_N, R_1, \ldots, R_N)
\]

**Lie group generalized-\( \alpha \) solver [B. & Cardona 2010]**

1. No rotation parameterization is explicitly involved
2. The solution inherently remains on the manifold
3. The integration formulae are nonlinear
4. The classical generalized-\( \alpha \) algorithm is a special case when \( G' \) is a linear vector space
5. Second-order accuracy + numerical damping
6. Much simpler than parameterization-based methods
Treatment of rotations: benchmark

Spinning top with spherical ellipsoid of inertia and constant follower torque

\[ \dot{\mathbf{R}} = \mathbf{R} \tilde{\mathbf{\Omega}} \]

\[ \mathbf{J} \dot{\mathbf{\Omega}} + \mathbf{\Omega} \times \mathbf{J} \mathbf{\Omega} = \mathbf{C} \]

⇒ analytical solution [Romano 2008]

\[
\text{error} = \frac{1}{n_e} \sum_{i=1}^{n_e} \| \mathbf{x}(t_i) - \mathbf{x}^{ref}(t_i) \|
\]

Controller dynamics

Coupled dynamic equations:

\[
\begin{align*}
\ddot{\mathbf{q}} &= \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}, t) \\
\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}, t)
\end{align*}
\]

\[
\begin{align*}
\mathbf{q}_{n+1} &= \mathbf{q}_n + h \dot{\mathbf{q}}_n + h^2 (0.5 - \beta) \mathbf{a}_n + h^2 \beta \mathbf{a}_{n+1} \\
\dot{\mathbf{q}}_{n+1} &= \dot{\mathbf{q}}_n + h (1 - \gamma) \mathbf{a}_n + h \gamma \mathbf{a}_{n+1} \\
\mathbf{x}_{n+1} &= \mathbf{x}_n + h (1 - \theta) \mathbf{w}_n + h \theta \mathbf{w}_{n+1}
\end{align*}
\]

\[
\begin{align*}
(1 - \alpha_m) \mathbf{a}_{n+1} + \alpha_m \mathbf{a}_n &= (1 - \alpha_f) \dot{\mathbf{q}}_{n+1} + \alpha_f \ddot{\mathbf{q}}_n \\
(1 - \delta_m) \mathbf{w}_{n+1} + \delta_m \mathbf{w}_n &= (1 - \delta_f) \dot{\mathbf{x}}_{n+1} + \delta_f \ddot{\mathbf{x}}_n
\end{align*}
\]

To be solved with:

\[
\begin{align*}
\ddot{\mathbf{q}}_{n+1} &= \mathbf{g}(\mathbf{q}_{n+1}, \dot{\mathbf{q}}_{n+1}, \mathbf{x}_{n+1}, t_{n+1}) \\
\ddot{\mathbf{x}}_{n+1} &= \mathbf{f}(\mathbf{q}_{n+1}, \dot{\mathbf{q}}_{n+1}, \mathbf{x}_{n+1}, t_{n+1})
\end{align*}
\]

Order conditions:

\[
\begin{align*}
\gamma &= 0.5 + \alpha_f - \alpha_m \\
\theta &= 0.5 + \delta_f - \delta_m
\end{align*}
\]
Summary

The generalized-α method combines

- Second-order accuracy (demonstrated for ODEs)
- Adjustable numerical damping
- Computational efficiency for large and stiff problems

Extension to coupled DAEs on Lie groups with a consistent treatment of:

- Kinematic constraints
- Rotational variables
- Control state variables

Outline

- Introduction
- Modelling of multibody & mechatronic systems
- Time integration algorithms
  - Topology optimization of structural components
    - Motivation
    - Method
    - Two-dofs robot arm
Motivation

Structural topology optimization: [Bendsøe & Kikuchi, 1988] [Sigmund, 2001]

- Design volume
- Material properties
- Boundary conditions
- Applied loads
- Objective function
- Design constraints

FE discretization & optimization

Large scale problem!

Motivation

Achievements in structural topology optimization
- Gradient-based algorithms (CONLIN, MMA, GCMMA)...
- Relevant problem formulations (SIMP penalization)...

A powerful design tool:

[Poncelet et al., 2005]
Motivation

Our objective: Topology optimization for the design of components of multibody systems

Equivalent static load approach, see e.g. [Kang & Park, 2005]

⇒ experience and intuition are required
⇒ optimal solution for a wrong problem!

Motivation

Topology optimization based on the actual dynamic response

⇒ Flexible multibody model (FE)
⇒ Time integrator (g-α)
⇒ Sensitivity analysis
⇒ Coupling with an optimizer

Advantages:
⇒ Systematic approach
⇒ More realistic objective function
Topology optimization

Parameterization of the topology: for each element,
- one density variable is defined $x = \rho / \rho_0$, $x \in [0, 1]$
- the Young modulus is computed according to the SIMP law

$$E = x^p E_0$$

Global optimization framework

Coupled industrial software
- OOFELIE (simulation and sensitivity analysis)
- CONLIN (gradient-based optimization) [Fleury 1989]

Efficient and reliable sensitivity analysis?
- Direct differentiation technique
Sensitivity analysis

For one design variable $x$, direct differentiation leads to

$$M\ddot{q} + C_q \dot{q} + K_q q + \Phi_q^T \lambda' + r_{,x} = 0$$

Integration of the sensitivities

- iteration matrix already computed and factorized
- one linear pseudo-load case for each design variable

Inertia forces $\propto \rho$

Elastic forces $\propto E$

$\Rightarrow$ Analytical expressions for $r_{,x}$

Importance of an efficient sensitivity analysis:

- Test problem with (only) 60 design variables
  
  $\Rightarrow$ Finite difference (61 simulations)  
  $\Rightarrow$ CPU time = 141 s

  $\Rightarrow$ Direct differentiation (1 extended simulation)  
  $\Rightarrow$ CPU time = 16 s

Moreover, the direct differentiation method leads to higher levels of accuracy

[B. & Eberhard 2008]
Two dofs robot arm

Initial structural universe of beams:

Point-to-point joint trajectory

Equivalent static case

Minimization of the compliance

subject to a volume constraint

Final design:
Minimization of the tip deflection

\[ \frac{1}{t_f} \int_0^{t_f} \|\mathbf{r} - \mathbf{r}_{\text{rigid}}\|^2 \, dt \]

subject to a volume constraint

\[ V_{(i)} \leq 0.4 V_{\text{full},(i)} \]

Final design:

---

**Summary**

- Topology optimization of mechanisms components
- Equivalent static load ⇒ multibody dynamics approach
  - flexible multibody simulation
  - sensitivity analysis
  - coupling with an optimization code
- Application to a two dofs robot arm with truss linkages
  - importance of problem formulation
- Perspectives
  - Complex 3D mechanisms
  - Mechatronic systems
Future work

Optimal control methods?

- inverse dynamics problems:
  identification of applied loads from kinematic data
- control-structure design optimization:
  the trajectory can be tuned for a given structural design
  the structural design can be tuned for a given trajectory

How to exploit our background for OC? [Bottasso et al, 2004]

- Indirect OC approach: an existing simulation framework cannot be reused easily
- Direct OC approach: reuse of an existing simulation framework is more feasible, but one needs to solve large NLP problems

Thank you for your attention!
Modelling of flexible multibody systems

Flexible beam element

- Timoshenko-type geometrically exact model
- Two nodes A and B
- Nodal translations \((x_A, x_B)\) and rotations \((R_A, R_B)\)
- Strain energy: bending, torsion, traction and shear
- Kinetic energy: translation and rotation

Hinge element

- Two nodes A (on body 1) and B (on body 2)
- Nodal translations \((x_A, x_B)\) and rotations \((R_A, R_B)\)
- 5 kinematic constraints

\[
\begin{align*}
x_A - x_B &= 0 \\
\mu_1'(R_A) \cdot \xi_3'(R_B) &= 0 \\
\mu_2'(R_A) \cdot \xi_3'(R_B) &= 0
\end{align*}
\]
Minimize the mean compliance:

\[ \frac{1}{t_f} \int_0^{t_f} \sum_{i=1}^{n_e} c(i) \, dt \]

subject to a volume constraint

\[ V(i) \leq 0.4 \, V_{full,(i)} \]

Final design:

Choice of the objective function

A one-element test-case:

Objective functions:

- Mean compliance
- Mean square tip deflection