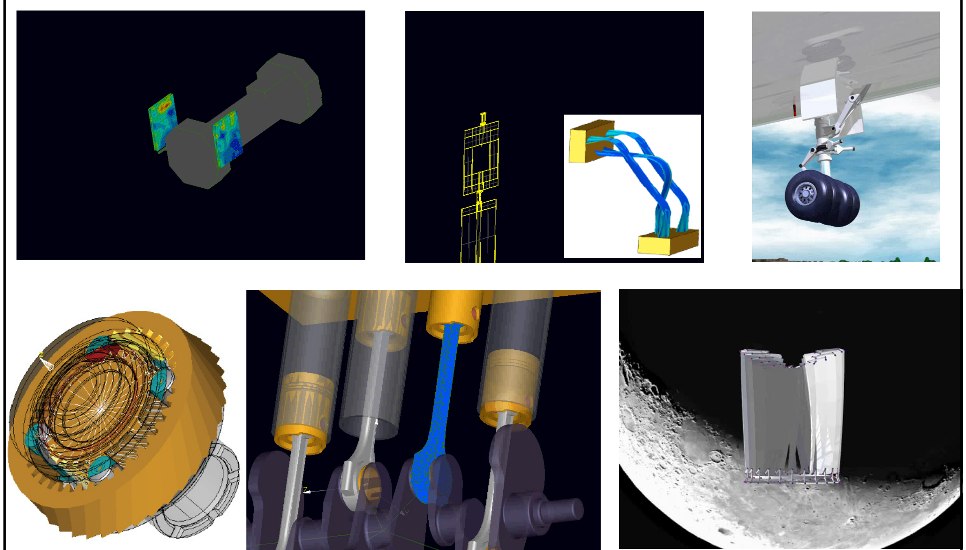
Numerical solution of DAEs in flexible multibody dynamics with applications in control and mechatronics

Olivier Brüls

Multibody & Mechatronic System Laboratory
Department of Aerospace and Mechanical Engineering
University of Liège, Belgium
o.bruls@ulg.ac.be

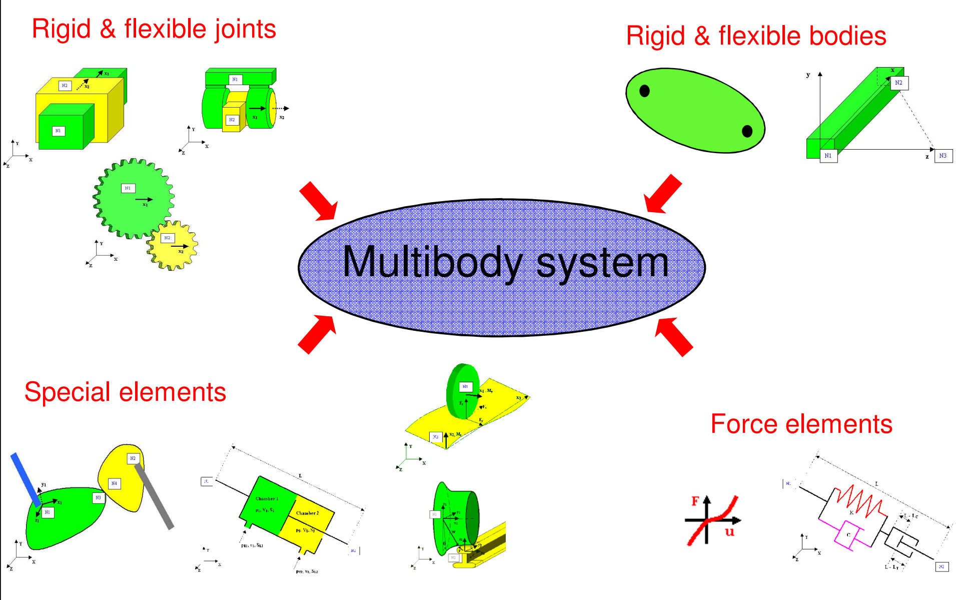
INRIA, Grenoble, October 21, 2010

Introduction



Some models developed using Samcef/Mecano

Finite element simulation library



Commercial simulation tools

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

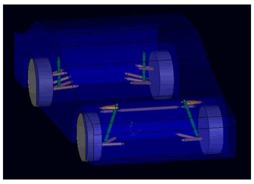
} Linear flexibility effects using the floating frame of reference method

Finite element approach
- SAMCEF MECANO
- OOFELIE

} General approach for nonlinear flexible systems

Integrated mechatronic design

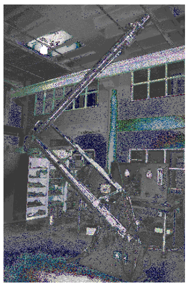
Active suspension



Mechanism | Actuator Sensor | Control

↓

Mechatronic design



Manipulator (Georgia Tech)

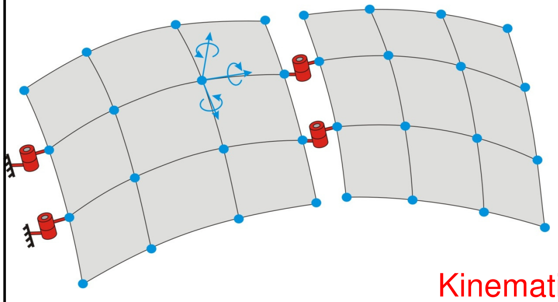
Integrated simulation software for mechatronic systems
⇒ relevant basis for the development of optimization tools

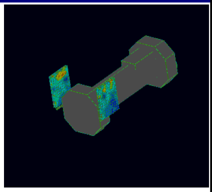
Outline

- ❑ Introduction
- ❑ Modelling of multibody & mechatronic systems
  - ➢ Modelling of flexible multibody systems
  - ➢ Modelling of coupled mechatronic systems
  - ➢ Application to a semi-active car suspension
  - ➢ Application to a wind turbine
- ❑ Time integration algorithms

Modelling of flexible multibody systems

Finite element approach [Géradin & Cardona 2001]





Absolute nodal coordinates

- translations & rotations
- geometric nonlinearities

Kinematic joints & rigidity conditions

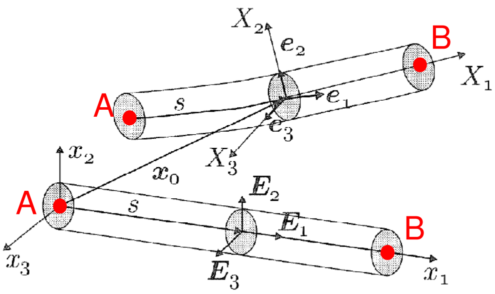
- algebraic constraints

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{g}_{gyr}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}_{int}(\mathbf{q}) + \mathbf{\Phi}_{\mathbf{q}}^T\boldsymbol{\lambda} &= \mathbf{g}_{ext} \\ \mathbf{\Phi}(\mathbf{q},t) &= \mathbf{0} \end{aligned}$$

index-3 DAE with rotation variables

Modelling of flexible multibody systems

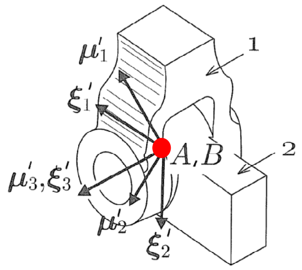
Flexible beam element



- Timoshenko-type geometrically exact model
- Two nodes A and B
- Nodal translations $(\mathbf{x}_A, \mathbf{x}_B)$ and rotations $(\mathbf{R}_A, \mathbf{R}_B)$
- Strain energy: bending, torsion, traction and shear
- Kinetic energy: translation and rotation

Modelling of flexible multibody systems

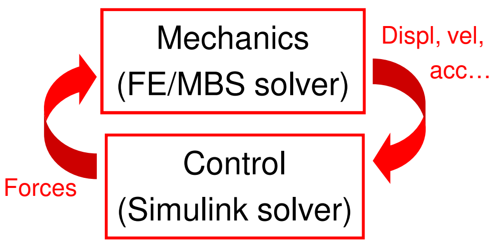
Hinge element



- Two nodes A (on body 1) and B (on body 2)
- Nodal translations $(\mathbf{x}_A, \mathbf{x}_B)$ and rotations $(\mathbf{R}_A, \mathbf{R}_B)$
- 5 kinematic constraints

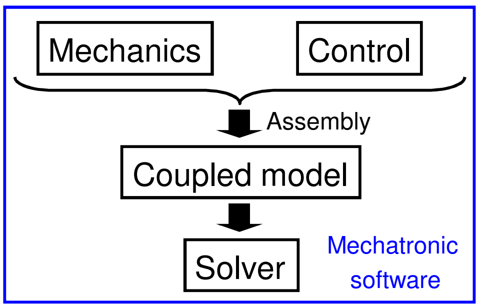
$$
\begin{aligned}
\mathbf{x}_A - \mathbf{x}_B &= \mathbf{0} \\
\boldsymbol{\mu}_1'(\mathbf{R}_A) \cdot \boldsymbol{\xi}_3'(\mathbf{R}_B) &= 0 \\
\boldsymbol{\mu}_2'(\mathbf{R}_A) \cdot \boldsymbol{\xi}_3'(\mathbf{R}_B) &= 0
\end{aligned}
$$

Simulation of coupled mechatronic systems



Co-simulation between 2 software packages

- Software interface
- Critical communication strategy between solvers

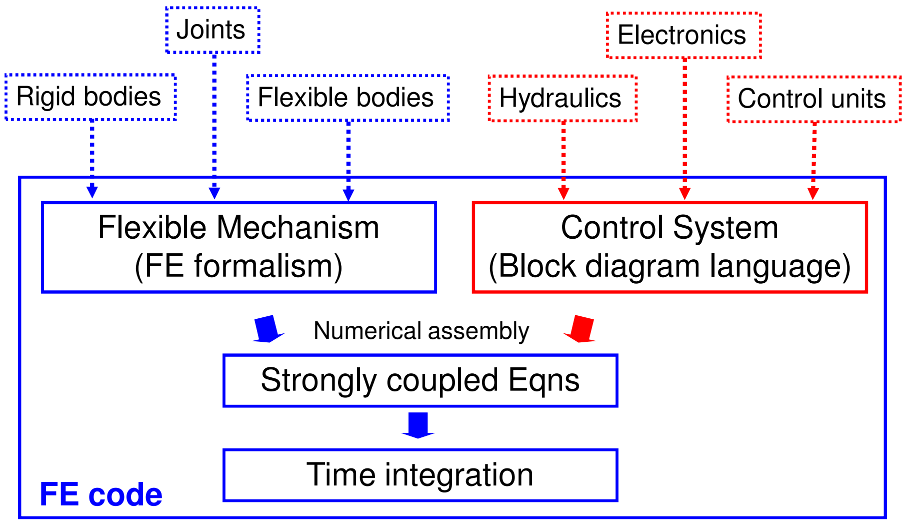


Monolithic approach

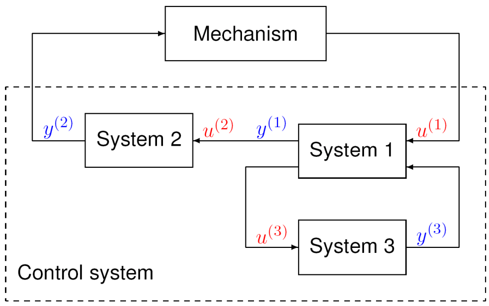
- Unified software:
  modelling + solver
- No software interface
- Strong (tight) coupling

Modelling of coupled mechatronic systems

Modular and monolithic FE approach



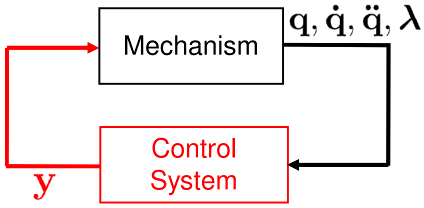
Modelling of coupled mechatronic systems



Block diagram language in a FE code

- Generic blocks : gain, integrator, transfer function…
  ⇒ "special" elements
- Control state/output variables ⇒ "special" dofs
- Interconnexions ⇒ variable sharing
- Numerical assembly according to the FE procedure

Modelling of coupled mechatronic systems



Coupled equations:

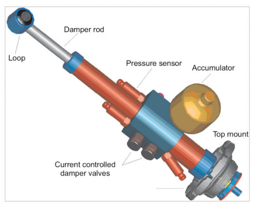
$$
\begin{aligned}
\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} &= \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, t) - \boldsymbol{\Phi}_{\mathbf{q}}^{T}\boldsymbol{\lambda} + \mathbf{L}\mathbf{y} \\
\mathbf{0} &= \boldsymbol{\Phi}(\mathbf{q}, t) \\
\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\lambda}, \mathbf{x}, \mathbf{y}, t) \\
\mathbf{y} &= \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\lambda}, \mathbf{x}, \mathbf{y}, t)
\end{aligned}
$$

Time-integration scheme for coupled 1st/2nd order DAE ?

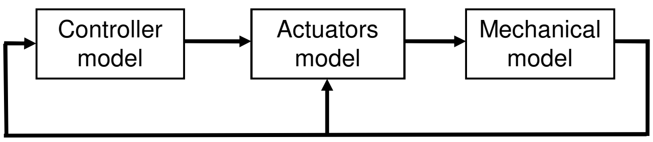
- Classical ODE solvers : multistep & Runge-Kutta methods
- Generalized-α time integration scheme

Semi-active car suspension

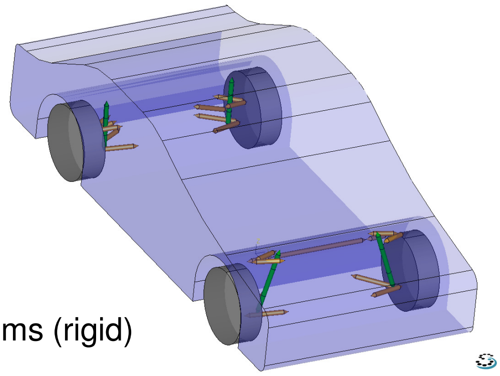
Work in collaboration with KULeuven-PMA and UCL-CEREM (PAI5/6)



- Hydraulic actuators with electrical valves
- Accelerometers on the car body



Semi-active suspension

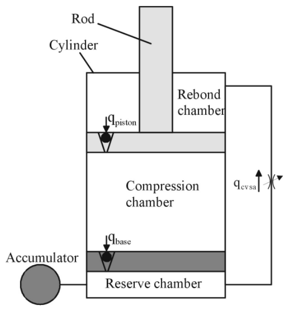


Mechanical model

- Car-body (rigid)
- Suspension mechanisms (rigid) and passive springs
- Slider-crank direction mechanism
- Wheel-ground contact

$\Rightarrow$ 600 dofs, but sparse matrices !

Semi-active suspension



Full dynamic model

- Valves: nonlinear relation flow / Δp
- Chambers: isentropic compression
- Available as C-functions

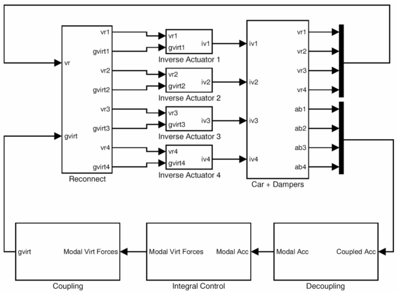
$\Rightarrow$

$$\begin{aligned}
\mathbf{u}^{(damp)} &= [l^r \quad v^r \quad i^v]^T \\
\mathbf{x}^{(damp)} &= [p^{reb} \quad p^{comp}]^T \\
\mathbf{y}^{(damp)} &= [g^a]^T
\end{aligned}$$

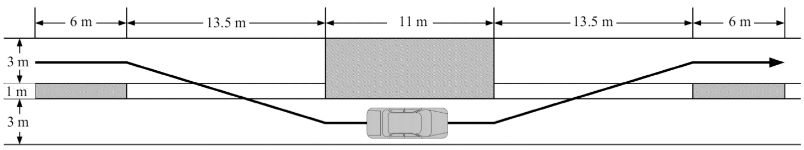
Semi-active suspension

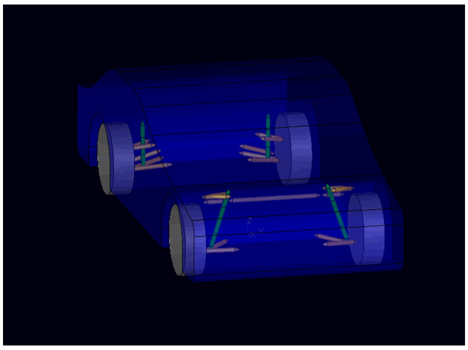
Controller [Lauwerys et al, 2004]

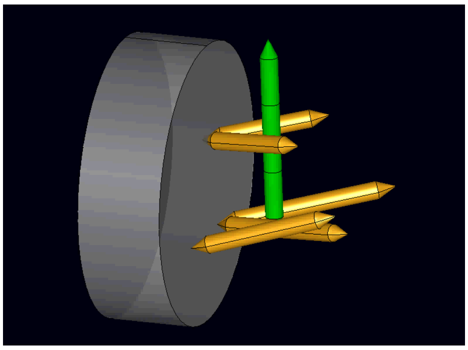
- Feedback linearization
  (Compensation of the actuator nonlinearity)
- Sky-hook modal controller (roll, pitch, heave)
- Block diagram model developed with Simulink



Semi-active car suspension

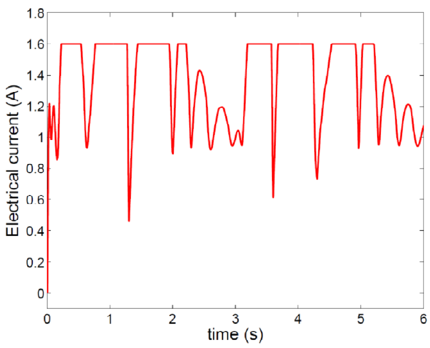




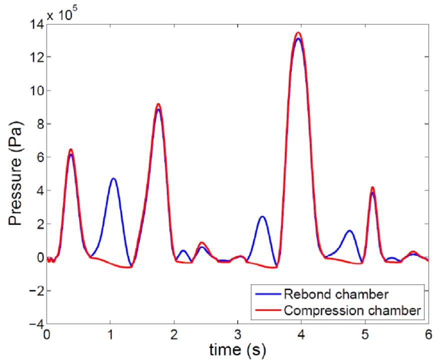


Semi-active car suspension

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



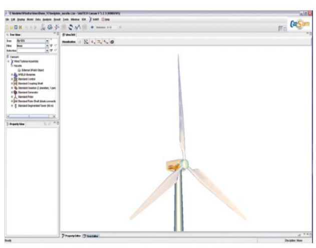
Hydraulic pressures (Pa)
(rear-left wheel)



Mechatronic modelling of wind turbines

Computer-aided analysis for wind turbines

- Existing software: GH Bladed, Simpack Wind etc.
- Importance of dynamic loads + amplification effects
- Samcef for Wind Turbines (S4WT) since 2004



GUI

Front view

Back view

(Courtesy: Samtech)

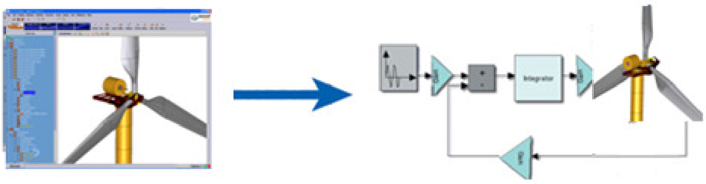
Mechatronic modelling of wind turbines

S4WT is based on SAMCEF/MECANO

- Aeroelastic beam model of the blades
- Drive-train models based on a dedicated gear element

Generator and control system: two modelling approaches

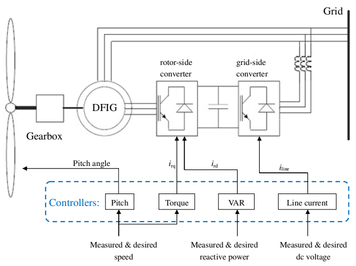
- weak coupling with a DLL exported from Simulink
- monolithic approach using control elements in Samcef



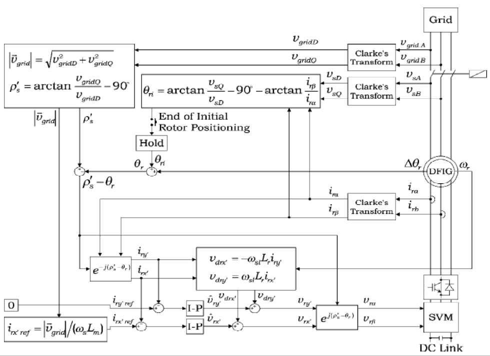
(Courtesy: Samtech)

Mechatronic modelling of wind turbines

DFIG generator model



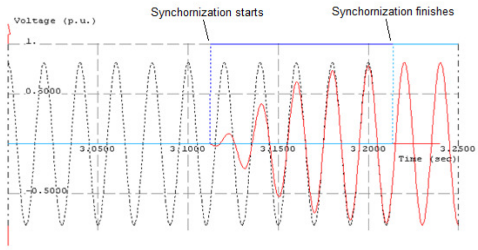
Vector control scheme and grid synchronization



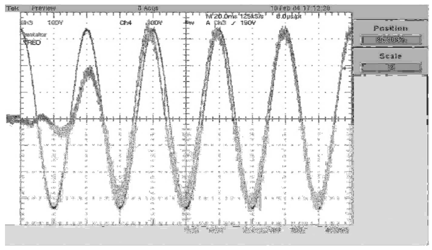
Tapia, G. et al. Methodology for smooth connection of doubly fed induction generators to the grid. *IEEE Transactions on Energy Conversion*, 24,4(2009), 959-971

Mechatronic modelling of wind turbines

Grid synchronization control results



Simulation on Samcef



Experiment from
Tapia et al

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-α time integrator is used to solve the strongly coupled problem

Liège - Belgium

Campus of Sart-Tilman

Liège-Guillemins train station

Outline

- ❑ Introduction
- ❑ Modelling of multibody & mechatronic systems
- ❑ Time integration algorithms
  - ➢ Generalized-α method
  - ➢ Kinematic constraints
  - ➢ Controller dynamics
  - ➢ Rotation variables

Generalized-α method

Numerical time-integration methods

- Standard integrators: multistep, Runge-Kutta
- Methods from structural dynamics (Newmark, HHT, g-α)
- Energy conserving/decaying 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, controller dynamics and rotation variables?

Generalized-α method

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

Newmark implicit formulae:

$$\begin{aligned} \mathbf{q}_{n+1} &= \mathbf{q}_n + h\dot{\mathbf{q}}_n + h^2(0.5-\beta)\mathbf{a}_n + h^2\beta\mathbf{a}_{n+1} \\ \dot{\mathbf{q}}_{n+1} &= \dot{\mathbf{q}}_n + h(1-\gamma)\mathbf{a}_n + h\gamma\mathbf{a}_{n+1} \end{aligned}$$

Generalized-α method [Chung & Hulbert, 1993]

$$(1-\alpha_m)\mathbf{a}_{n+1} + \alpha_m\mathbf{a}_n = (1-\alpha_f)\ddot{\mathbf{q}}_{n+1} + \alpha_f\ddot{\mathbf{q}}_n$$

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

- Two kinds of acceleration variables: $\mathbf{a}_n \neq \ddot{\mathbf{q}}_n$
- Algorithmic parameters: $\gamma,\ \beta,\ \alpha_f,\ \alpha_m$
  2nd order accuracy & numerical damping

Kinematic constraints

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} &= \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, t) - \boldsymbol{\Phi}_{\mathbf{q}}^T \boldsymbol{\lambda} \\ \mathbf{0} &= \boldsymbol{\Phi}(\mathbf{q}, t) \end{aligned}$$

Direct integration of the index-3 DAE problem using g-α

- 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]

Controller dynamics

Coupled dynamic equations: $\begin{cases} \ddot{\mathbf{q}} = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}, t) \\ \dot{\mathbf{x}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}, t) \end{cases}$

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

$$\begin{cases} (1-\alpha_m)\mathbf{a}_{n+1} + \alpha_m\mathbf{a}_n = (1-\alpha_f)\ddot{\mathbf{q}}_{n+1} + \alpha_f\ddot{\mathbf{q}}_n \\ (1-\delta_m)\mathbf{w}_{n+1} + \delta_m\mathbf{w}_n = (1-\delta_f)\dot{\mathbf{x}}_{n+1} + \delta_f\dot{\mathbf{x}}_n \end{cases}$$

To be solved with : $\begin{cases} \ddot{\mathbf{q}}_{n+1} = \mathbf{g}(\mathbf{q}_{n+1}, \dot{\mathbf{q}}_{n+1}, \mathbf{x}_{n+1}, t_{n+1}) \\ \dot{\mathbf{x}}_{n+1} = \mathbf{f}(\mathbf{q}_{n+1}, \dot{\mathbf{q}}_{n+1}, \mathbf{x}_{n+1}, t_{n+1}) \end{cases}$

Order conditions: $\begin{cases} \gamma = 0.5 + \alpha_f - \alpha_m \\ \theta = 0.5 + \delta_f - \delta_m \end{cases}$

Rotation variables

Rotation variables at the core of the FE method for flexible MBS

- Orientation of a rigid body
- Orientations of nodes at a joint
- Orientation of beam cross-sections & shell director vectors

$$\dot{\mathbf{R}} = \mathbf{R}\widetilde{\boldsymbol{\Omega}}$$
$$\mathbf{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} - \mathbf{C}(\mathbf{R}, t) = \mathbf{0}$$

ODE on a Lie group

Rotation parameters, e.g. Cartesian rotation vector $\boldsymbol{\psi}$

$$\mathbf{R} = \mathbf{I}_3 + \frac{\sin\phi}{\phi}\widetilde{\boldsymbol{\psi}} + \frac{1-\cos\phi}{\phi^2}\widetilde{\boldsymbol{\psi}}\widetilde{\boldsymbol{\psi}} \quad \text{where} \quad \phi = \|\boldsymbol{\psi}\|$$

$$\mathbf{M}(\boldsymbol{\psi})\ddot{\boldsymbol{\psi}} + \mathbf{g}(\boldsymbol{\psi}, \dot{\boldsymbol{\psi}}, t) = \mathbf{0}$$

ODE on a vector space

Rotation parameterization difficulties

- Complexity of parameterized equations of motion
- Singularities of minimal parameterizations

How to avoid parameterization singularities?

- Redundant parameterization + kinematic constraints [Betsch & Steinmann 2001]
- Rotationless formulation, e.g. ANCF [Shabana]
- Minimal parameterization + updated Lagrangian point of view [Cardona & Géradin 1989]
- Lie group time integrator without *a priori* parameterization [Simo & Vu-Quoc 1988; B., Cardona & Arnold 2010]

We are interested in simplified codes

Can we avoid rotation parameters?

- Can we solve the non-parameterized form of the equations of motion? (which only involves $\mathbf{R}$ and $\mathbf{\Omega}$)

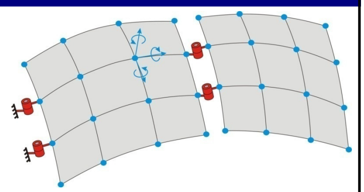
- Standard time integrators work for ODEs/DAEs on a vector space, but not for equations on a Lie group

- Lie group integration methods can solve ODEs on a Lie group [Crouch & Grossman 1993; Munthe-Kaas 1995,1998] and also [Simo & Vu-Quoc 1988; Bottasso & Borri 1998]

We study a method to solve ODEs and DAEs on Lie groups for MBS, based on the generalized-α scheme

Configuration space as a Lie group

Nodal configuration variables



$$q = (\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{R}_1, \ldots, \mathbf{R}_N)$$

with $\mathbf{x}_i \in \mathbb{R}^3, \mathbf{R}_i \in SO(3)$

$$SO(3) = \{\mathbf{R} : \mathbb{R}^3 \to \mathbb{R}^3 \mid \mathbf{R}^T\mathbf{R} = \mathbf{I}_3, \det \mathbf{R} = +1\}$$

The configuration evolves on the k-dimensional Lie group

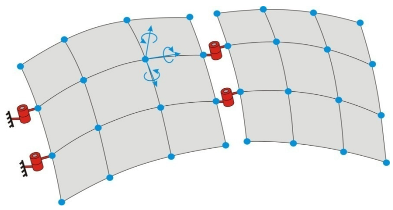
$$G = \mathbb{R}^3 \times \ldots \times \mathbb{R}^3 \times SO(3) \times \ldots \times SO(3)$$

with the composition $q_{tot} = q_1 \circ q_2$ such that

$$\mathbf{x}_{i,tot} = \mathbf{x}_{i,1} + \mathbf{x}_{i,2}$$

$$\mathbf{R}_{i,tot} = \mathbf{R}_{i,1}\mathbf{R}_{i,2}$$

Constrained equations of motion



Joints and rigidity conditions ⇒ *m* kinematic constraints $\mathbf{\Phi}(q)$

⇒ Submanifold of dimension *k-m*

$$N = \{q \in G : \mathbf{\Phi}(q) = \mathbf{0}\}$$

Semi-discretized equations of motion (DAE on a Lie group)

$$\begin{aligned} \dot{q} &= DL_q(e) \cdot \widetilde{\mathbf{v}} \\ \mathbf{M}(q)\dot{\mathbf{v}} + \mathbf{g}(q, \mathbf{v}, t) + \mathbf{B}^T(q)\boldsymbol{\lambda} &= \mathbf{0} \\ \mathbf{\Phi}(q) &= \mathbf{0} \end{aligned}$$

Lie group generalized-α method

$$\begin{aligned} \mathbf{M}(q_{n+1})\dot{\mathbf{v}}_{n+1} &= -\mathbf{g}(q_{n+1}, \mathbf{v}_{n+1}, t_{n+1}) - \mathbf{B}^T(q_{n+1})\boldsymbol{\lambda}_{n+1} \\ \mathbf{\Phi}(q_{n+1}) &= \mathbf{0} \\ \mathbf{\Delta q}_n &= \mathbf{v}_n + (0.5 - \beta)h^2\mathbf{a}_n + \beta h^2 \mathbf{a}_{n+1} \\ q_{n+1} &= q_n \circ \exp\left(h\widetilde{\mathbf{\Delta q}_n}\right) \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + (1-\gamma)h\mathbf{a}_n + \gamma h \mathbf{a}_{n+1} \\ (1-\alpha_m)\mathbf{a}_{n+1} + \alpha_m \mathbf{a}_n &= (1-\alpha_f)\dot{\mathbf{v}}_{n+1} + \alpha_f \dot{\mathbf{v}}_n \end{aligned}$$

1. Non-parameterized equations of motion at time n+1
2. Nonlinear integration formulae (composition & exponential)
3. For a vector space ⇒ classical generalized-α algorithm
4. Newton iterations involve k+m unknowns [B. & Cardona 2010]

$$\mathbf{S}_t = \begin{bmatrix} \beta'\mathbf{M} + \gamma'\mathbf{C}_t + \mathbf{K_t T} & \mathbf{B}^T \\ \mathbf{BT} & \mathbf{0} \end{bmatrix}$$ + h-scaling

Analytical expression of the exponential map

Single translation system : $\exp(\widetilde{\mathbf{x}}) = \mathbf{x}$

Single rotation system :

$$\exp(\widetilde{\mathbf{x}}) = \mathbf{I}_3 + \frac{\sin\phi}{\phi}\widetilde{\mathbf{x}} + \frac{1-\cos\phi}{\phi^2}\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}$$

where $\phi = \|\mathbf{x}\|$

The scheme by [Simo & Vu-Quoc,1988] is a special case when $\alpha_m = \alpha_f = 0$

General case : component-wise definition $\Rightarrow$ numerical effort scales linearly with the number of rotational variables

Proof of second-order accuracy for DAEs

- Local and global errors analysis in the Lie algebra
- Baker-Campbell-Hausdorff (BCH) formula

$$\exp(\widetilde{\mathbf{v}}_1) \circ \exp(\widetilde{\mathbf{v}}_2) = \exp(\widetilde{\mathbf{v}}_1 + \widetilde{\mathbf{v}}_2 + \mathcal{O}(h^r))$$

- Local errors: $\hat{q}_{n+1} = q(t_{n+1}) \circ \exp(\widetilde{\mathbf{l}}_n^q)$
- Magnus expansion of the exact solution

$$q(t_{n+1}) = q(t_n) \circ \exp(h\widetilde{\boldsymbol{\nu}})$$

$$h\widetilde{\boldsymbol{\nu}} = h\widetilde{\mathbf{v}}(t_n) + \frac{h^2}{2}\dot{\widetilde{\mathbf{v}}}(t_n) + \frac{h^3}{6}\ddot{\widetilde{\mathbf{v}}}(t_n) + \frac{h^3}{12}[\widetilde{\mathbf{v}}(t_n), \dot{\widetilde{\mathbf{v}}}(t_n)] + \mathcal{O}(h^4)$$

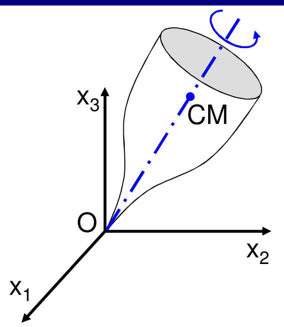
- Global error recursion: see multistep methods for DAEs with higher order error terms from the BCH formula

Global errors are $O(h^2)$ for fixed step-sizes in all components if

$$\gamma = 0.5 + \alpha_f - \alpha_m \ ,$$

$$\alpha_m < \alpha_f < 0.5 \quad \text{and} \quad \beta > 0.25 + (\alpha_f - \alpha_m)/2$$

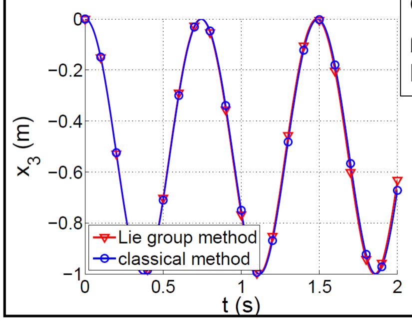
Benchmark 1: Spinning top



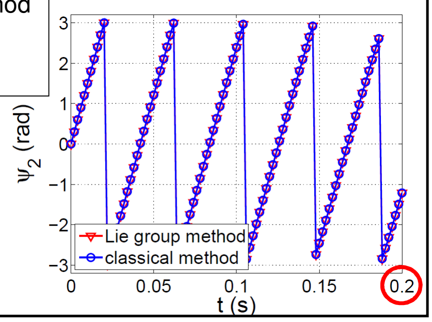
Nodal coordinates of the CM: $q = (\mathbf{x}, \mathbf{R})$

$$\dot{\mathbf{R}} = \mathbf{R}\widetilde{\boldsymbol{\Omega}}$$

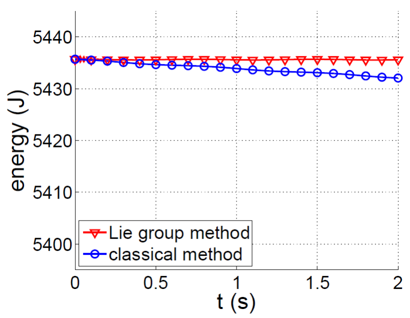
$$\begin{aligned} m\ddot{\mathbf{x}} - \boldsymbol{\lambda} &= m\boldsymbol{\gamma} \\ \mathbf{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} + \widetilde{\mathbf{X}}\mathbf{R}^T\boldsymbol{\lambda} &= \mathbf{0} \\ -\mathbf{x} + \mathbf{R}\mathbf{X} &= \mathbf{0} \end{aligned}$$

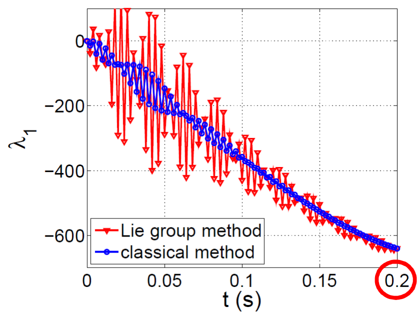


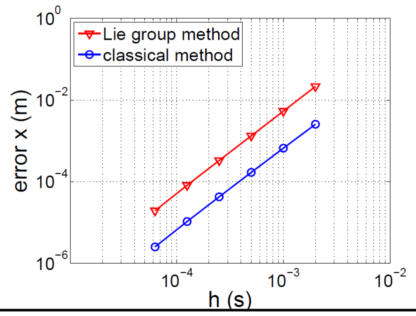
Gen-α method
ρ = 0.9
h = 0.002 s

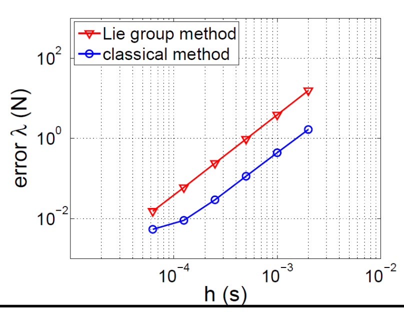


Benchmark 1: Spinning top

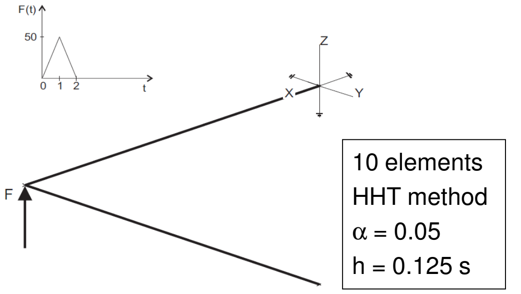


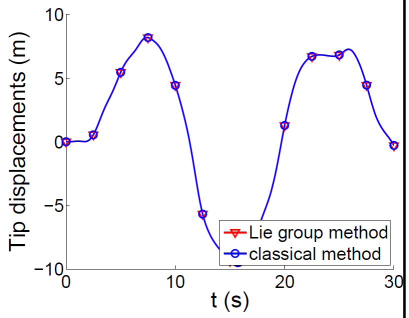


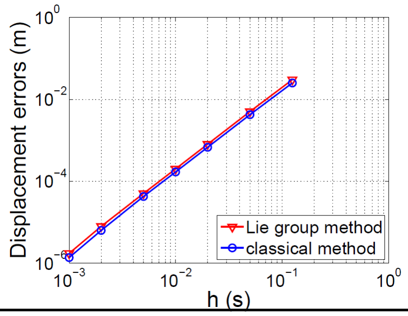


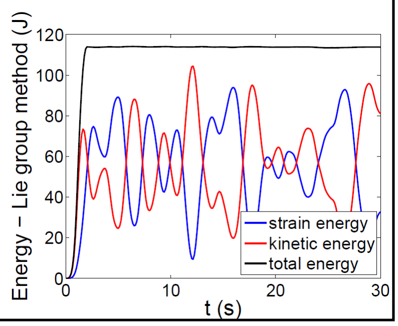


Benchmark 2: Rightangle flexible beam









Summary

The generalized-α integration 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 and simplified treatment of:

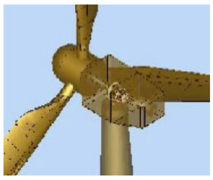
- Kinematic constraints
- Rotational variables in SO(3)
- Control state variables

It is a promising approach for the analysis, control & optimization of large scale flexible multibody systems

Current challenges

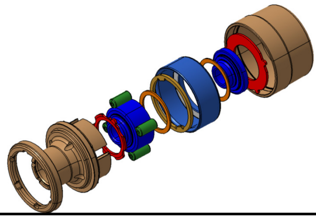
Transmission lines

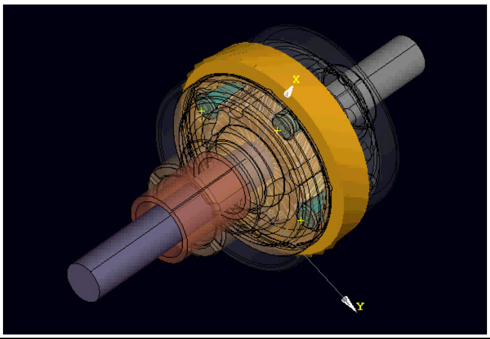
- ➢Gear pairs
- ➢Unilateral contact with high stiffness
- ➢Friction
- ➢Backlash



Regularization, event-driven or time stepping approaches?

Torsen type-C differential



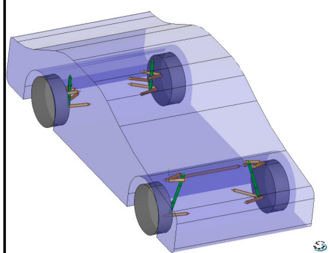


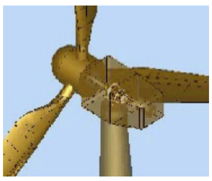
Thank you for your attention!

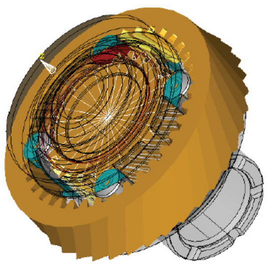
Numerical solution of DAEs in flexible multibody dynamics with applications in control and mechatronics

Olivier Brüls

Acknowledgements: M. Arnold, A. Cardona, Q.Z. Chen, P. Duysinx, M. Géradin, J.C. Golinval, G. Virlez







Université de Liège

INRIA, Grenoble, October 21, 2010