Sensitivity analysis for flexible multibody systems formulated on a Lie group

Olivier Brüls, Valentin Sonneville

Department of Aerospace and Mechanical Engineering

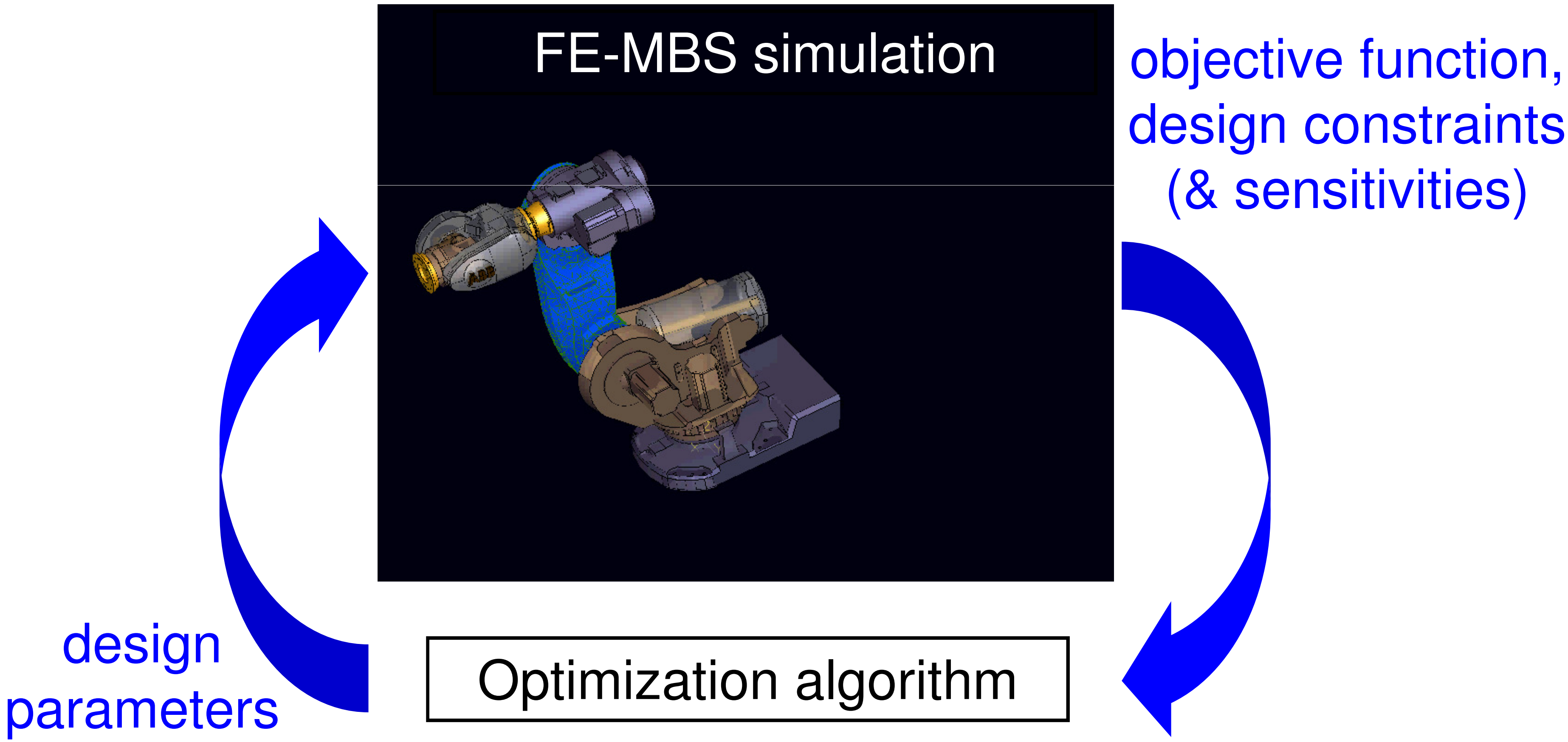
University of Liège, Belgium

EUROMECH Colloquium 524

Enschede, February 29, 2012

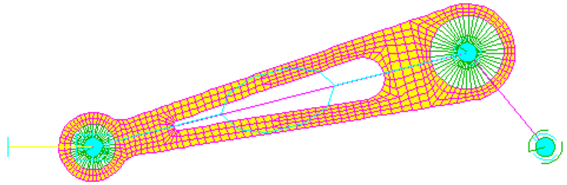
FE-based MBS optimization

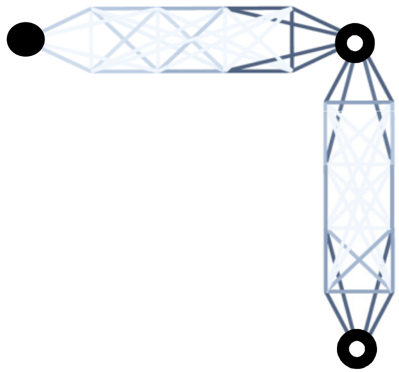
- Deformations, vibration levels & stresses are available
- Integrated simulation of flexible MBS (simplicity of workflow)



FE-based MBS optimization

Structural optimization



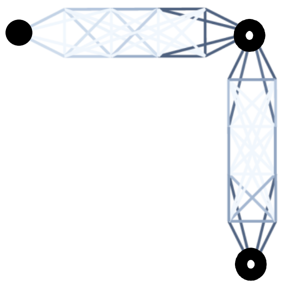


Output y

Input u?

Inverse dynamics of flexible MBS using optimal control methods

FE-based MBS optimization



Problem	Design variables	Cost function & constraints
Density-based topology optimization	• Densities in each element of the mesh	• Mean compliance • Mean tip deflection •… • Stresses in each element of the mesh at each time step

Large scale optimization problems

➢Gradient-based methods (SQP, IP, CONLIN, MMA, etc)

➢Sparse implementation

➡ Efficient evaluation of **sensitivities** is essential

Methods for sensitivity analysis

High cost of finite differences for large scale problems

- n_p additional simulations for fwd/bwd differences (order 1)
- $2\ n_p$ additional simulations for central differences (order 2)

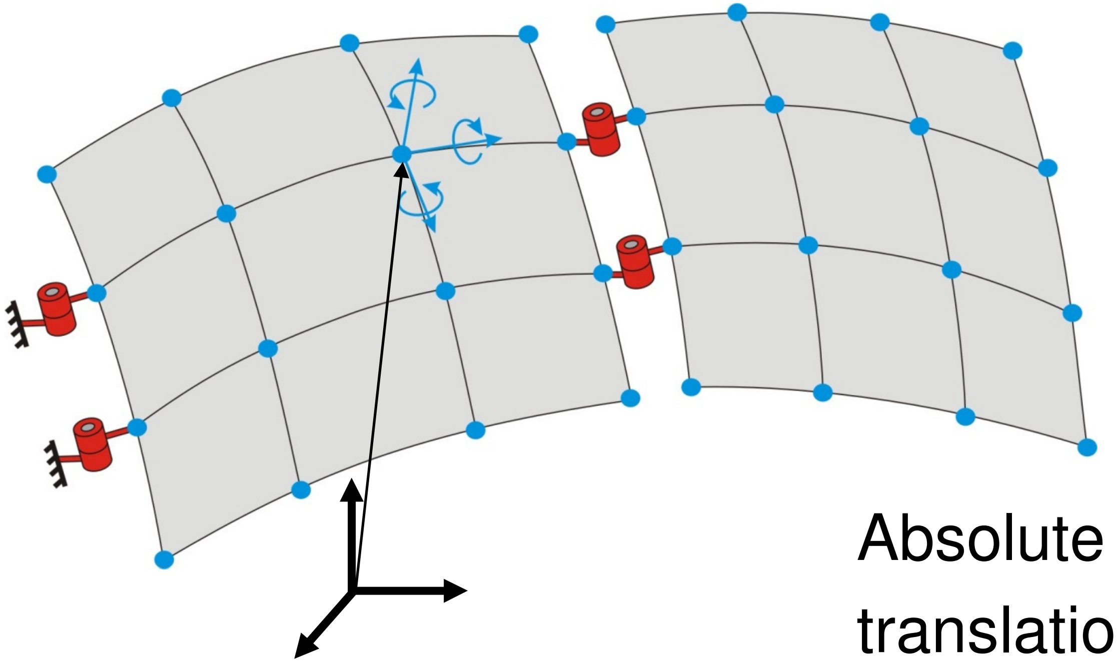
Automatic differentiation

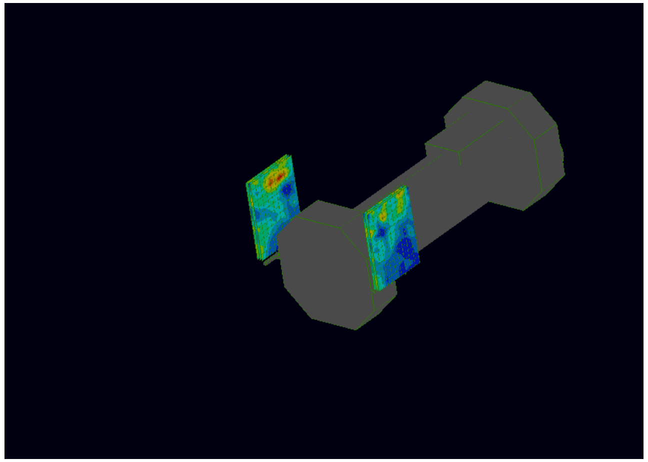
- High reliability but suboptimal code (unnecessary operations need to be removed manually)
- Maintenance difficulties for an evolving simulation code

Semi-analytical methods (direct differentiation / adjoint variable)

- Optimized but manual implementation
- Strong amplification of the intricacy of a simulation code
- Feasible for flexible MBS?

Classical FE approach for flexible MBS





Absolute nodal coordinates:

translations & **rotation parameters**

Kinematic joints & rigidity conditions $\Rightarrow$ algebraic constraints

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

Successful for simulation codes but challenging for SA!

Interest in simpler parameterization-free approache

Outline

1. FE-based optimization
2. Lie group formulations and solvers
3. Sensitivity analysis on a Lie group
4. Numerical example
5. Conclusion

Lie group formulation

The configuration of a MBS can be described as an element of a matrix Lie group (parameterization-free approach).

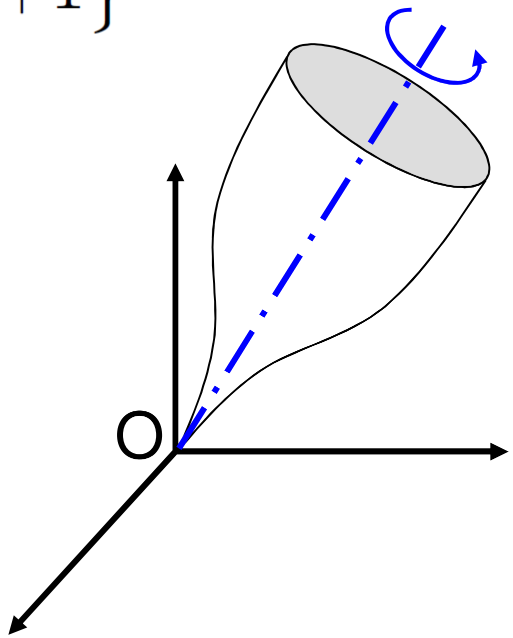
Example: $\mathbf{R}(t) \in SO(3)$

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3\times 3} | \mathbf{R}^T\mathbf{R} = \mathbf{I}_3, \det\mathbf{R} = +1 \right\}$$

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

$$\mathbf{\Omega} = [\Omega_1 \; \Omega_2 \; \Omega_3]^T$$

$$\widetilde{\mathbf{\Omega}} = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix}$$



A Lie group is not a linear space!

Lie group formulation

Index-3 DAE on a Lie group

$$\begin{aligned} \dot{q} &= q\tilde{\mathbf{v}} \\ \mathbf{M}\dot{\mathbf{v}} - \hat{\mathbf{v}}^T\mathbf{M}\mathbf{v} + \mathbf{g}(q,t) + \mathbf{B}^T(q)\boldsymbol{\lambda} &= \mathbf{0} \\ \boldsymbol{\Phi}(q) &= \mathbf{0} \end{aligned}$$

- The configuration is described by the matrix q
- The velocity is described by a vector $\mathbf{v}$, related with the matrix $\tilde{\mathbf{v}}$
- The mass matrix is constant
- Parameterization-free formulation!

Lie group time integrator

Solution of DAEs on a Lie group [B. and Cardona 2010]

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

- Inspired by Newmark / generalized-α methods
- Analytical form of the exponential map
- Newton iterations for vector unknowns (not matrix unknowns)
- Second-order convergence [B. et al 2011]
- Reduced-index formulation [Arnold et al 2011]

Outline

1. FE-based optimization
2. Lie group formulations and solvers
3. Sensitivity analysis on a Lie group
4. Numerical example
5. Conclusion

Sensitivity analysis on a Lie group

Let us consider a single design parameter p

$$\begin{aligned} \dot{q} &= q\tilde{\mathbf{v}} \\ \mathbf{M}(p)\dot{\mathbf{v}} - \hat{\mathbf{v}}^T\mathbf{M}(p)\mathbf{v} + \mathbf{g}(q,p,t) + \mathbf{B}^T(q,p)\boldsymbol{\lambda} &= \mathbf{0} \\ \boldsymbol{\Phi}(q,p) &= \mathbf{0} \end{aligned}$$

and a single criterion function

$$\Psi(p) = G(q(t_f), \mathbf{v}(t_f), p) + \int_{t_0}^{t_f} F(q, \mathbf{v}, \dot{\mathbf{v}}, \boldsymbol{\lambda}, p)\ dt$$

Extension to several parameters and criteria is straightforward

Sensitivity analysis on a Lie group

The velocity vector **v** was defined as $\dot{q} = q\widetilde{\mathbf{v}}$

and it represents the derivative of q w.r.t. the time.

Recall the $SO(3)$ example: $\dot{\mathbf{R}} = \mathbf{R}\widetilde{\mathbf{\Omega}}$

Likewise, if $()'$ denotes a derivative w.r.t. p,

the sensitivity vector $\mathbf{w}$ is defined as

$$q' = q\tilde{\mathbf{w}}$$

and it represents the derivative of q w.r.t. the parameter.

As $\mathbf{v}$, the vector $\mathbf{w}$ belongs to a linear space.

Sensitivity analysis on a Lie group

$$\Psi(p) = G(q(t_f), \mathbf{v}(t_f), p) + \int_{t_0}^{t_f} F(q, \mathbf{v}, \dot{\mathbf{v}}, \boldsymbol{\lambda}, p)\, dt$$

d/dp

$$\frac{d\Psi}{dp} = \left(G_q \boxed{\mathbf{w}} + \frac{\partial G}{\partial \mathbf{v}} \boxed{\mathbf{v}'} + \frac{\partial G}{\partial p} \right)_{t_f} + \int_{t_0}^{t_f} \left(F_q \boxed{\mathbf{w}} + \frac{\partial F}{\partial \dot{\mathbf{v}}} \boxed{\dot{\mathbf{v}}'} + \frac{\partial F}{\partial \mathbf{v}} \boxed{\mathbf{v}'} + \frac{\partial F}{\partial \boldsymbol{\lambda}} \boxed{\boldsymbol{\lambda}'} + \frac{\partial F}{\partial p} \right) dt$$

With the definition of G_q and F_q

$$\begin{aligned} D_1 G(q, \mathbf{v}, p) \cdot (q\tilde{\mathbf{w}}) &= G_q \mathbf{w} \\ D_1 F(q, \mathbf{v}, \dot{\mathbf{v}}, \boldsymbol{\lambda}, p) \cdot (q\tilde{\mathbf{w}}) &= F_q \mathbf{w} \end{aligned}$$

Second-order derivatives do not commute: $\dot{\mathbf{w}} = \mathbf{v}' - \tilde{\mathbf{v}}\mathbf{w}$

Sensitivity analysis on a Lie group

$$
\begin{aligned}
\dot{q} &= q\tilde{\mathbf{v}} \\
\mathbf{M}(p)\dot{\mathbf{v}} - \hat{\mathbf{v}}^T\mathbf{M}(p)\mathbf{v} + \mathbf{g}(q,p,t) + \mathbf{B}^T(q,p)\boldsymbol{\lambda} &= \mathbf{0} \\
\boldsymbol{\Phi}(q,p) &= \mathbf{0}
\end{aligned}
$$

d/dp ➡

$$
\boxed{
\begin{aligned}
\dot{\mathbf{w}} &= \mathbf{v}' - \tilde{\mathbf{v}}\mathbf{w} \\
\mathbf{M}\dot{\mathbf{v}}' + \mathbf{C}_t\mathbf{v}' + \mathbf{K}_t\mathbf{w} + \mathbf{B}^T\boldsymbol{\lambda}' &= -\mathbf{res}' \\
\mathbf{B}\mathbf{w} &= -\boldsymbol{\Phi}'
\end{aligned}
}
$$

with the pseudo-loads:

$$
\begin{aligned}
\mathbf{res}' &= (\partial\mathbf{M}/\partial p)\dot{\mathbf{v}} - \hat{\mathbf{v}}(\partial\mathbf{M}/\partial p)\mathbf{v} + (\partial\mathbf{g}/\partial p) + (\partial\mathbf{B}/\partial p)^T\boldsymbol{\lambda} \\
\boldsymbol{\Phi}' &= \partial\boldsymbol{\Phi}/\partial p
\end{aligned}
$$

Linear 1st order DAE for $\mathbf{w}$ and $\mathbf{v}'$

- Classical DAE time integration methods can be used
- The generalized-α method does not apply as such
- Parameterization-free framework!

Direct differentiation of the time integrator

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

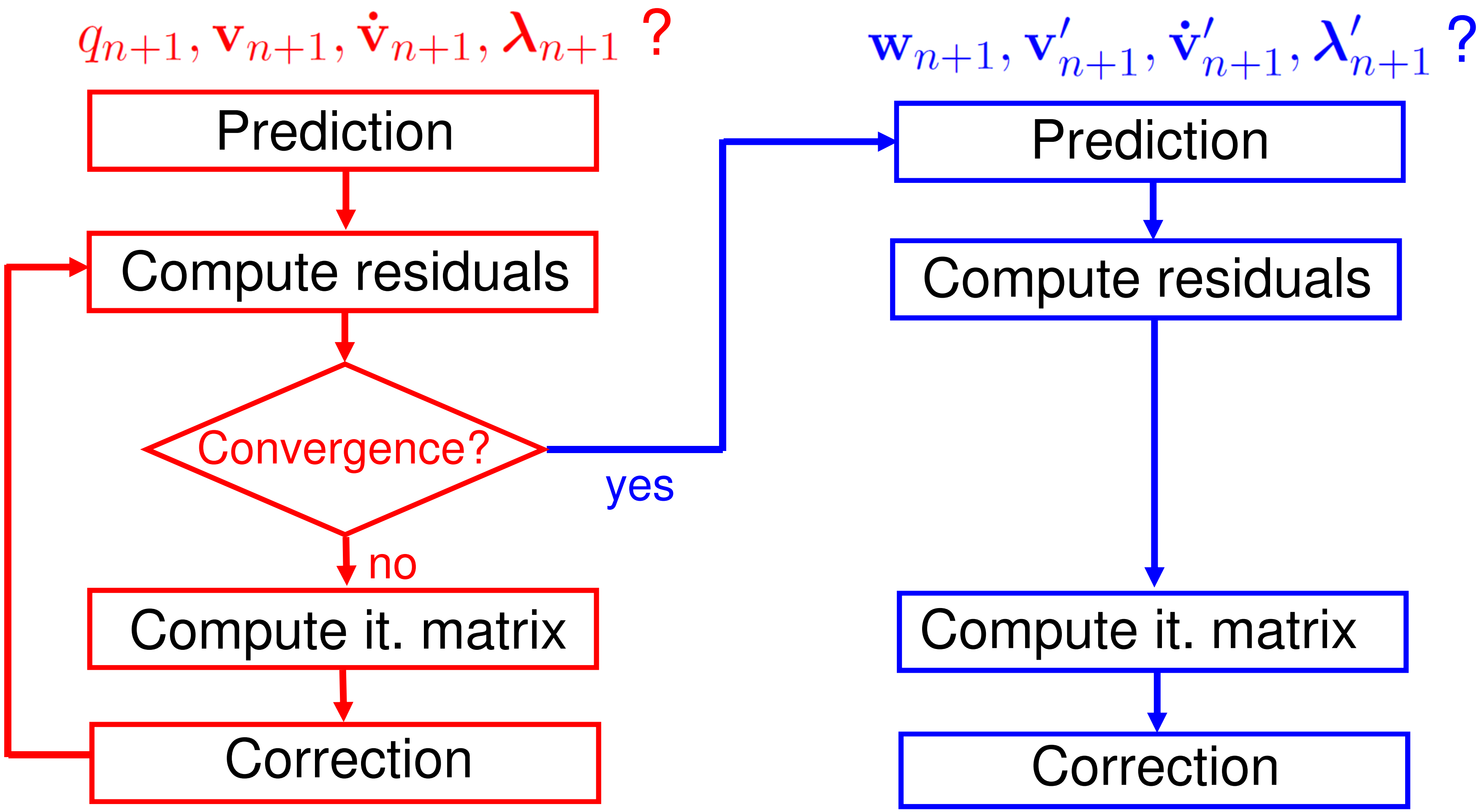
d/dp

$$
\begin{aligned}
\mathbf{M}\dot{\mathbf{v}}'_{n+1} + \mathbf{C}_t\mathbf{v}'_{n+1} + \mathbf{K}_t\mathbf{w}_{n+1} &= -\mathbf{res}' - \mathbf{B}^T\boldsymbol{\lambda}'_{n+1} \\
\mathbf{B}\mathbf{w}_{n+1} &= -\boldsymbol{\Phi}' \\
\mathbf{w}_{n+1} &= \mathbf{A}(\Delta\mathbf{x}_{n+1})\mathbf{w}_n + \mathbf{T}(\Delta\mathbf{x}_{n+1})\Delta\mathbf{x}'_{n+1} \\
\Delta\mathbf{x}'_{n+1} &= h\mathbf{v}'_n + (0.5-\beta)h^2\mathbf{a}'_n + \beta h^2\mathbf{a}'_{n+1} \\
\mathbf{v}'_{n+1} &= \mathbf{v}'_n + (1-\gamma)h\mathbf{a}'_n + \gamma h \mathbf{a}'_{n+1} \\
(1-\alpha_m)\mathbf{a}'_{n+1} + \alpha_m\mathbf{a}'_n &= (1-\alpha_f)\dot{\mathbf{v}}'_{n+1} + \alpha_f \dot{\mathbf{v}}'_n
\end{aligned}
$$

Linear algebraic equations for the sensitivities

- Same « iteration » matrix as for the nominal problem
- Pseudo-loads have to be evaluated (analytically or by FD)
- One transient linear load case for each design variable, regardless of the number of design criteria

Direct differentiation of the time integrator



Adjoint variable method

Augmented criterion with one adjoint variable per constraint:

$$\delta\Psi = \delta\Psi + \boldsymbol{\pi}^T\delta\boldsymbol{\zeta} + \boldsymbol{\rho}^T\delta\boldsymbol{\chi} + \int_{t_0}^{t_f} \left(\boldsymbol{\mu}^T\delta\mathbf{r}(q,\mathbf{v},\dot{\mathbf{v}},\boldsymbol{\lambda},t) + \boldsymbol{\nu}^T\delta\boldsymbol{\Phi}(q)\right)\,dt$$

$$\begin{aligned}
\boldsymbol{\pi} : \boldsymbol{\zeta} &= q(t_0,p) - q_0(p) = \mathbf{0} \\
\boldsymbol{\rho} : \boldsymbol{\chi} &= \mathbf{v}(t_0,p) - \mathbf{v}_0(p) = \mathbf{0} \\
\boldsymbol{\mu}(t) : \mathbf{r} &= \mathbf{M}(p)\dot{\mathbf{v}} - \hat{\mathbf{v}}^T\mathbf{M}(p)\mathbf{v} + \mathbf{g}(q,p,t) + \mathbf{B}^T(q,p)\boldsymbol{\lambda} = \mathbf{0} \\
\boldsymbol{\nu}(t) : \boldsymbol{\Phi} &= \boldsymbol{\Phi}(q,p) = \mathbf{0}
\end{aligned}$$

The adjoint formulation is obtained after integration by part…

Adjoint variable method

$$\delta\Psi = (G_p + \boldsymbol{\rho}^T \boldsymbol{\chi}_p + \boldsymbol{\pi}^T \boldsymbol{\zeta}_p)\delta p + \int_{t_0}^{t_f} (F_p + \boldsymbol{\mu}^T \mathbf{r}_p + \boldsymbol{\nu}^T \boldsymbol{\Phi}_p)\delta p \, dt$$

provided that the adjoint variables satisfy

$$\begin{aligned}
\mathbf{M}\ddot{\boldsymbol{\mu}} - (\mathbf{M}\hat{\mathbf{v}} + \mathbf{C}_t)^T \dot{\boldsymbol{\mu}} + (\mathbf{K}_t + \mathbf{C}_t\hat{\mathbf{v}} - \dot{\mathbf{C}}_t)^T \boldsymbol{\mu} + \mathbf{B}^T \boldsymbol{\nu} &= (-\frac{d^2}{dt^2} F_{\mathbf{M}} + \frac{d}{dt} F_{\mathbf{C}} - F_{\mathbf{K}})^T \\
\mathbf{B}\boldsymbol{\mu} &= -F_{\boldsymbol{\lambda}}^T \\
\mathbf{r}_{\mathbf{M}}^T \boldsymbol{\mu}(t_f) &= -(F_{\mathbf{M}} + G_{\mathbf{C}})_{t_f}^T \\
\mathbf{r}_{\mathbf{M}}^T \dot{\boldsymbol{\mu}}(t_f) &= (F_{\mathbf{C}} + \boldsymbol{\mu}^T \mathbf{r}_{\mathbf{C}} - \frac{d}{dt} F_{\mathbf{M}} + G_{\mathbf{K}})_{t_f}^T \\
\boldsymbol{\chi}_{\mathbf{C}}^T \boldsymbol{\rho} &= (F_{\mathbf{M}} + \boldsymbol{\mu}^T \mathbf{r}_{\mathbf{M}})_{t_0}^T \\
\boldsymbol{\zeta}_{\mathbf{K}}^T \boldsymbol{\pi} &= (F_{\mathbf{C}} + \boldsymbol{\mu}^T \mathbf{r}_{\mathbf{c}} - \frac{d}{dt} F_{\mathbf{M}} - \dot{\boldsymbol{\mu}}^T \mathbf{r}_{\mathbf{M}} - \boldsymbol{\rho}^T \boldsymbol{\chi}_{\mathbf{K}})_{t_0}^T
\end{aligned}$$

Linear 2nd order DAE for μ and ν, which can be solved backward in time using the classical generalized-α method

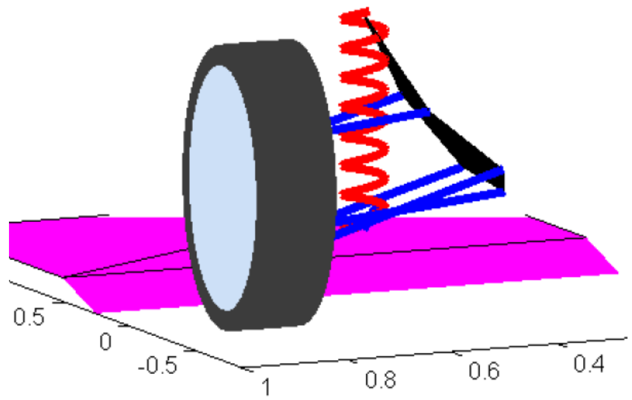
- « Iteration matrix » is related to the original problem
- Pseudo-loads have to be evaluated (analytically or by FD)
- One transient linear load case for each design criterion (regardless of the number of design variable)

Outline

1. FE-based optimization
2. Lie group formulations and solvers
3. Sensitivity analysis on a Lie group
4. Numerical example
5. Conclusion

Numerical example

Quarter-car suspension passing over a bump



- Design parameters:

  p_1= stiffness coefficient

  p_2 = damping coefficient

- Design criteria

$$\Psi_0 = \int_{t_0}^{t_f} \dot{v}_{z,chassis}^2(t)\ dt$$

One objective function

$$\Psi_1(t) = F_{ground}(t) - F_0 \leq 0$$

One constraint / time step

with $F_{ground}(t) = k_{wheel} \Delta z_{wheel}$

$$\Psi_2(t) = d_0 - \sqrt{(\mathbf{x}_{spring,chassis}(t) - \mathbf{x}_{spring,bar}(t))^2} \leq 0$$

One constraint / time step

Numerical example: pseudo loads

In the adjoint variable method, the pseudo load associated with

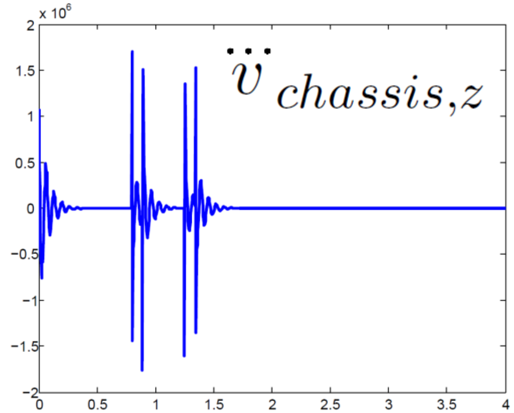
$$\Psi_0 = \int_{t_0}^{t_f} \dot{v}^2_{z,chassis}(t)\ dt$$

is computed as $(-\frac{d^2}{dt^2}F_{\mathbf{M}} + \frac{d}{dt}F_{\mathbf{C}} - F_{\mathbf{K}})^T = -2\dddot{v}_{chassis,z}$

One algebraic variable u and one algebraic constraint are introduced in the equations of motion

$$u = \dot{v}_{chassis,z}$$

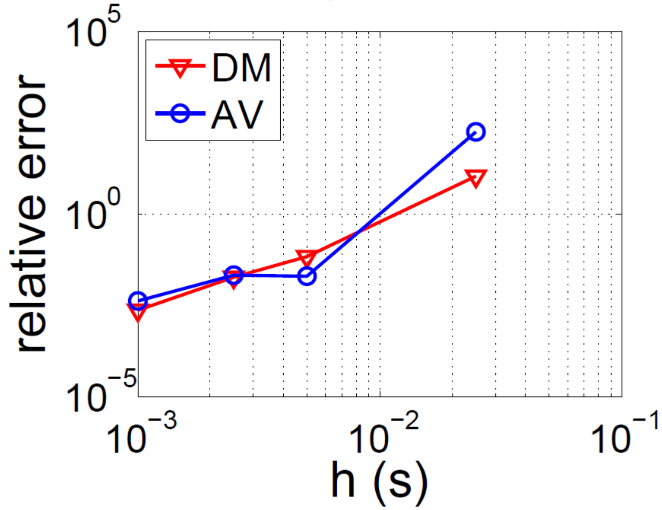
and the generalized-α formulae are used to solve for $\dot{u} = \ddot{v}_{chassis,z}$ and $\ddot{u} = \dddot{v}_{chassis,z}$



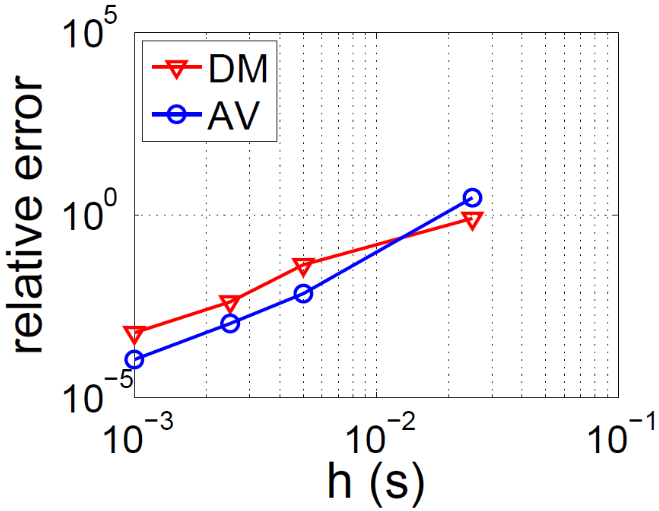
Any better idea?

Numerical example: sensitivities

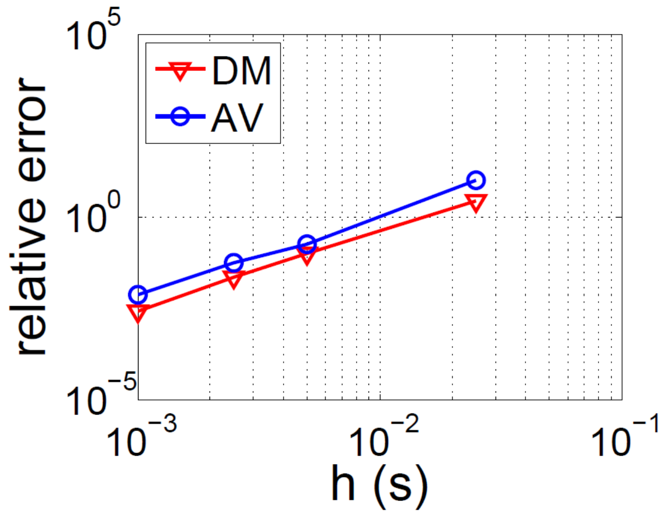
$d\Psi_0/dp_1$



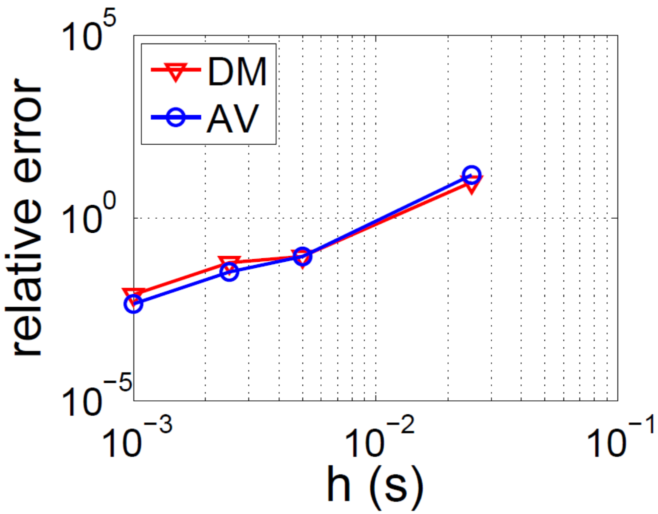
$d\Psi_1/dp_1$



$d\Psi_0/dp_2$

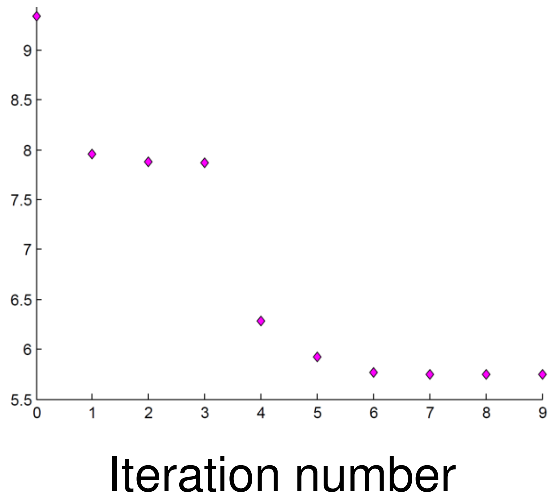


$d\Psi_1/dp_2$

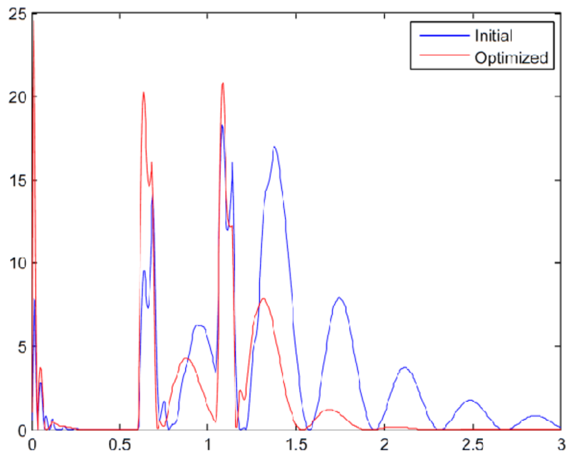


Numerical example: optimization

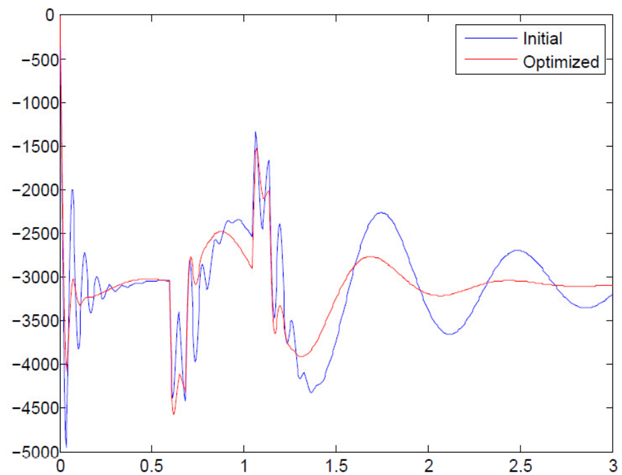
Ψ_0



$\dot{v}^2_{z,chassis}(t)$



$\Psi_1(t) = F_{ground}(t) - F_0$



Ψ_1 and Ψ_2 are imposed at each time step, however, only the gradient of active constraints has to be evaluated

- Direct differentiation: weakly affected by the number of criteria
- Adjoint variable: number of linear backward time integrations is equal to the number of active constraints

Conclusion

FE-based MBS optimization

- Intricate sensitivity analysis for « classical » formalisms
- Lie group methods $\Rightarrow$ simpler parameterization-free formulations & solvers

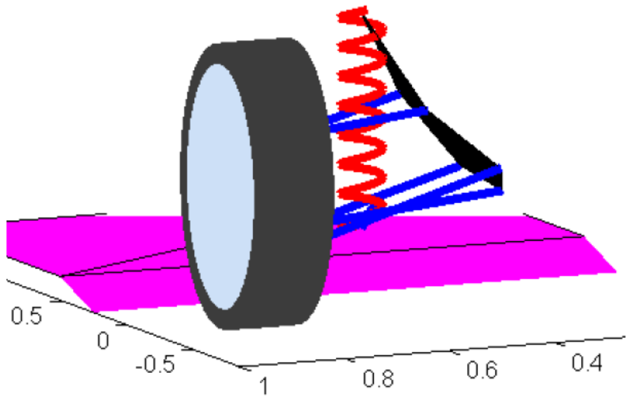
Sensitivities in a parameterization-free Lie group framework

- Direct differentiation vs. adjoint variable method
- One linear load case per design variable or per criterion
- Large parts of the simulation code can be reused
- Pseudo-loads can be numerically sensitive in the AVM

Quarter-car suspension example

- 2nd-order convergence in time is observed in most cases
- Fast convergence of gradient-based optimization

Thank you for your attention!



Sensitivity analysis for flexible multibody systems formulated on a Lie group

Olivier Brüls, Valentin Sonneville

Université de Liège

EUROMECH Colloquium 524
Enschede, February 29, 2012

LTAS

Overview of Lie group integration methods

Local (incremental) parameterization of the equations of motion

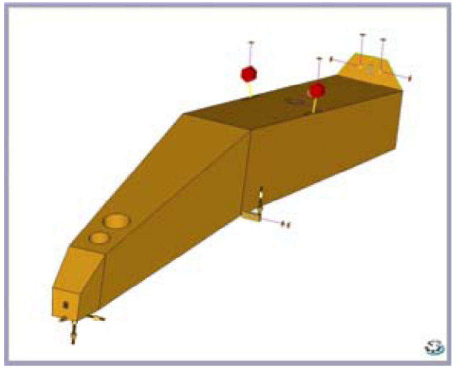
- Cardona & Géradin (1989): HHT method for flexible MBS
- Munthe-Kaas (1995, 1998): RK method for ODEs
- Bottasso & Borri (1998): RK and EC methods for flexible MBS

Integration formulae on a Lie group using the exponential map

- Simo (1988, 1991): Newmark and EC scheme for nonlinear structures
- Crouch & Grossman (1993): RK and multistep methods for ODEs
- B. et al (2010, 2011): Generalized-α method for flexible MBS

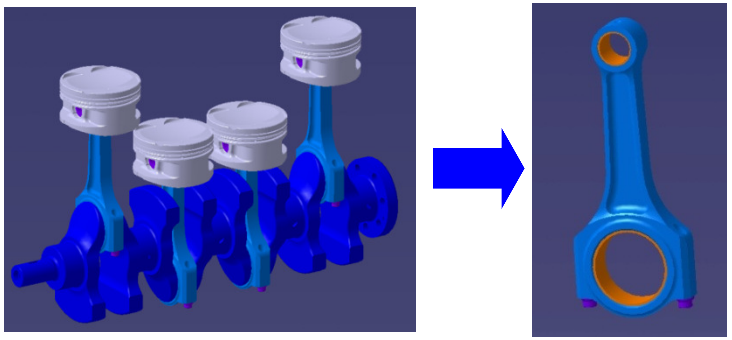
Motivation: structural optimization

Static structures

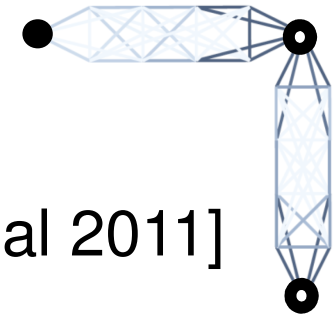


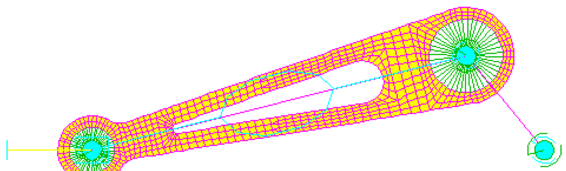
Components in mechanisms?

- Equivalent static-load approach [Kang & Park 2005]



- FE-MBS approach [B. et al 2007, Tromme et al 2011]





Motivation: optimal control

Inverse dynamics problem:

Find the control inputs u(t) leading to a given output motion y(t)

Solution strategies for underactuated systems

- Forward integration [Blajer and Kolodziejczyk 2004]
- Stable inversion [Seifried and Eberhard 2009]
- Optimal control [Bottasso et al 2004]

Flexible MBS are often

- non-minimum phase
- Difficult to study analytically

➡ FE-based optimal control [Bastos et al 2011]

Output y

Input u

Exact treatment of large rotations

Updated Lagrangian strategy [Cardona & Géradin, 1989]

$$\mathbf{R}(t_{n+1}) = \mathbf{R}(t_n)\, \mathbf{R}_{inc}(t_{n+1})$$

- Only the incremental rotation needs to be parameterized
- Geometrically exact and singularity-free approach
- Equivalent to a reparameterization at each time step

Implementation involves close links between

- the time integration scheme
- the rotation parameterization formulae
- the FE discretization
- the equations of motion

Successful for simulation codes but challenging for SA!

We need simpler parameterization-free approaches…