REAL-TIME METHODS IN FLEXIBLE MULTIBODY DYNAMICS

A thesis submitted for the degree of Doctor Ingeniero Industrial

Urbano Lugrís Armesto

University of La Coruña

Ferrol, November 2008

Outline

Outline

Introduction

Formulation in Relative Coordinates

Inertia Shape Integrals

Geometric Stiffening

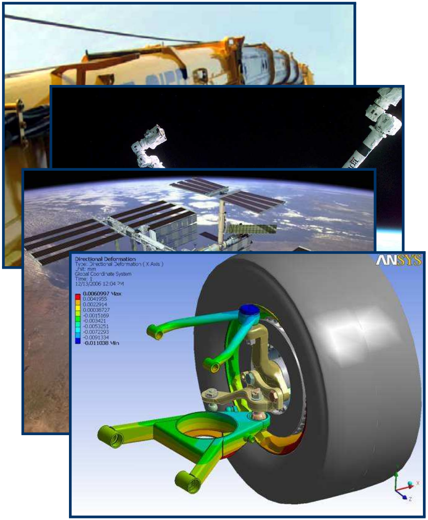
Conclusions and Future Research

Motivation (I)

- Our group has developed very efficient and robust formulations for the real–time simulation of rigid multibody systems

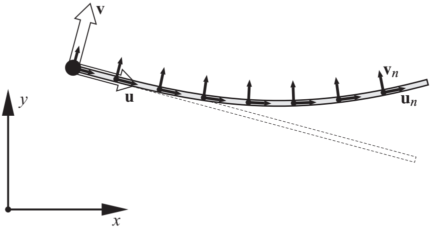
Motivation (II)

- Objective: include flexibility in real–time applications
  - Simulators, virtual reality. . .
- Many multibody applications cannot neglect flexibility
  - Slender components
  - Newer lightweight materials
  - High operational speed
- Flexible bodies require a higher computational effort
  - Elastic forces
  - Variable mass matrix



Existing Flexible MBS Approaches

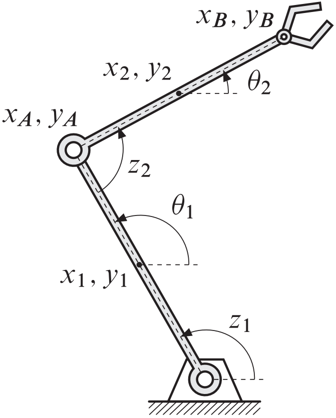
- Inertial frame
  - One inertial frame common to all the bodies in the system
  - J.C. Simó, L. Vu–Quoc, A. Cardona, M. Géradin, A.A. Shabana
- Floating frame: most efficient
  - One reference frame attached to each flexible body
  - E.J. Haug, A.A. Shabana, R.A. Wehage
- Corotational frame
  - Each finite element has a local frame of reference
  - T. Belytschko, B.J. Hsieh



Reference Coordinates in FFR Methods

Reference Coordinates ≡ Rigid Body Coordinates

- Reference point coordinates
  - Position and orientation in Cartesian coordinates
- Natural coordinates
  - Fully Cartesian
  - Points and unit vectors
- Relative coordinates
  - $O(n)$ fully–recursive formulations
  - $O(n^3)$ semi–recursive formulations



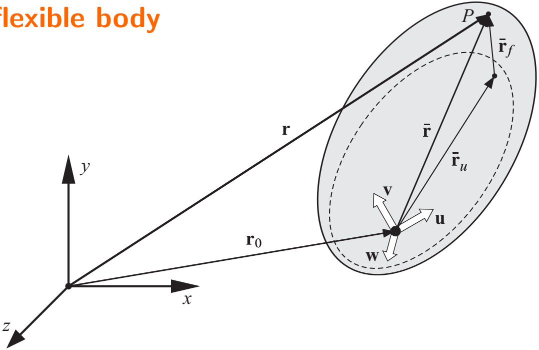
Objectives and Scope of the Present Work

- Development of a semi–recursive $O\left(n^3\right)$ FFR formulation
  - Based on an existing rigid–body one
- Comparison between natural and relative coordinates
  - Same comparison has been previously carried out in the rigid case
  - FFR formulation in natural coordinates as a reference
  - Both formulations share the same flexible body modeling
- Optimization of the inertia terms
  - Inertia Shape Integrals preprocessing
  - Implement in both absolute and relative coordinates
- Extension to nonlinear problems
  - Implementation and comparison of three techniques for capturing geometric stiffening in beams
  - Substructuring, Nonlinear stiffness matrix and Foreshortening

Outline

Flexible Body Modeling

General flexible body

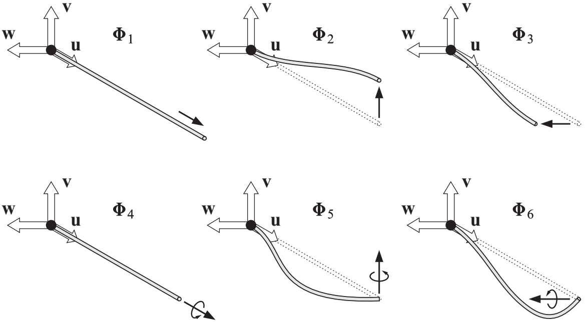


- Position of a point: $\mathbf{r} = \mathbf{r}_0 + \mathbf{A}\left(\bar{\mathbf{r}}_u + \bar{\mathbf{r}}_f\right); \quad \mathbf{A} = \begin{bmatrix}\mathbf{u} & \mathbf{v} & \mathbf{w}\end{bmatrix}$
- Elastic displacement in local coordinates (Craig–Bampton)

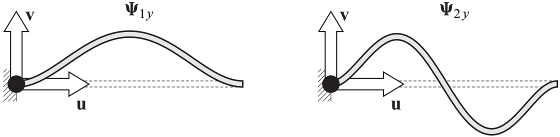
$$\bar{\mathbf{r}}_f = \sum_{i=1}^{n_s} \mathbf{\Phi}_i \eta_i + \sum_{j=1}^{n_d} \mathbf{\Psi}_j \xi_j = \mathbf{X}\mathbf{y}$$

Craig–Bampton Reduction

Static modes: unit displacements at boundaries



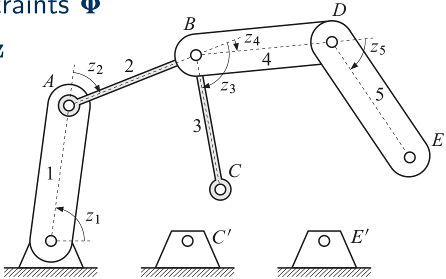
Dynamic modes: normal eigenmodes with fixed boundaries



Recursive Kinematics

- Closed loops are cut $\Longrightarrow$ constraints $\boldsymbol{\Phi}$
- Dependent relative coordinates $\mathbf{z}$
- Intermediate dynamic terms in Cartesian coordinates: $\bar{\mathbf{M}}, \bar{\mathbf{Q}}$
- Cartesian coordinates defined at velocity level (reference)

$$\mathbf{Z}^{\mathsf{T}} = \left\{ \dot{\mathbf{s}}^{\mathsf{T}} \quad \boldsymbol{\omega}^{\mathsf{T}} \quad \dot{\mathbf{y}}^{\mathsf{T}} \right\}$$

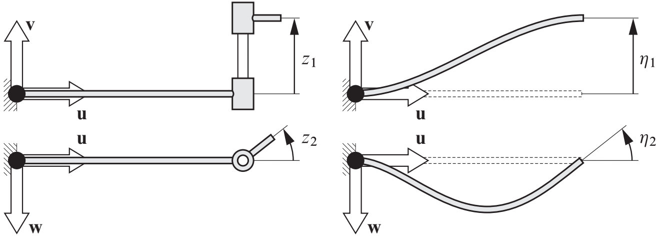


- Recursive relationships for velocities and accelerations

$$\left.\begin{aligned} \mathbf{Z}_{rj} &= \mathbf{Z}_{ri} + \mathbf{b}_j \dot{z}_j \\ \dot{\mathbf{Z}}_{rj} &= \dot{\mathbf{Z}}_{ri} + \mathbf{b}_j \ddot{z}_j + \mathbf{d}_j \end{aligned}\right\} \Longrightarrow \mathbf{Z} = \mathbf{R}\dot{\mathbf{z}}$$

Projection of the Dynamic Terms

- Static modes behave analogously as kinematic joints



- Kinematic relations include now joints and static modes

$$\left.\begin{aligned} \mathbf{Z}_{rj} &= \mathbf{Z}_{ri} + \mathbf{b}_j \dot{z}_j + \boldsymbol{\varphi}_j^P \dot{\eta}_j^P \\ \dot{\mathbf{Z}}_{rj} &= \dot{\mathbf{Z}}_{ri} + \mathbf{b}_j \ddot{z}_j + \boldsymbol{\varphi}_j^P \ddot{\eta}_j^P + \mathbf{d}_j + \boldsymbol{\gamma}_j^P \end{aligned}\right\} \Longrightarrow \mathbf{Z} = \mathbf{R}\dot{\mathbf{z}}$$

- Projection into $\mathbf{z}$: $\boxed{\mathbf{M} = \mathbf{R}^\mathsf{T}\bar{\mathbf{M}}\mathbf{R}; \quad \mathbf{Q} = \mathbf{R}^\mathsf{T}\left(\bar{\mathbf{Q}} - \bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right)}$

Calculation of the Inertia Terms

- Corotational approximation

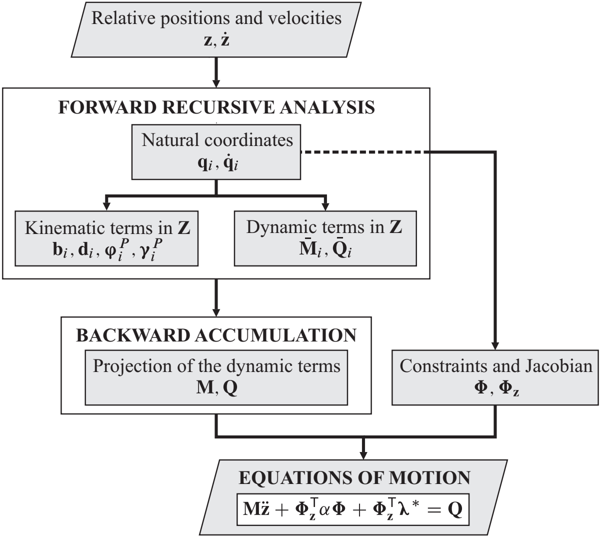
$$T = \frac{1}{2}\int_V \dot{\mathbf{r}}^{\mathsf{T}}\dot{\mathbf{r}}\,dm = \frac{1}{2}\dot{\mathbf{r}}^{*\mathsf{T}}\left(\int_V \mathbf{N}^{\mathsf{T}}\mathbf{N}\,dm\right)\dot{\mathbf{r}}^{*} = \frac{1}{2}\dot{\mathbf{r}}^{*\mathsf{T}}\mathbf{M}^{*}\dot{\mathbf{r}}^{*}$$

- Transformation matrix $\mathbf{B}$, assembled for the whole body

$$\mathbf{B}^{*} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\\ \dots\\ \mathbf{B}_n\end{bmatrix} = \begin{bmatrix}\mathbf{I}_3 & -\tilde{\mathbf{r}}_1 & \mathbf{A}\mathbf{X}_1\\ \mathbf{I}_3 & -\tilde{\mathbf{r}}_2 & \mathbf{A}\mathbf{X}_2\\ \dots & \dots & \dots\\ \mathbf{I}_3 & -\tilde{\mathbf{r}}_n & \mathbf{A}\mathbf{X}_n\end{bmatrix} \Longrightarrow \dot{\mathbf{r}}^{*} = \mathbf{B}^{*}\mathbf{Z}$$

- Projection of the finite element mass matrix: $\boxed{\bar{\mathbf{M}} = \mathbf{B}^{*\mathsf{T}}\mathbf{M}^{*}\mathbf{B}^{*}}$
- Velocity–dependent inertia forces: $\bar{\mathbf{Q}}_v = -\mathbf{B}^{*\mathsf{T}}\mathbf{M}^{*}\dot{\mathbf{B}}^{*}\mathbf{Z}$

Assembly of the Equations of Motion



Dynamic Formulation and Numerical Integration

- Index–3 Augmented Lagrangian

$$\boxed{\begin{aligned} &\mathbf{M}\ddot{\mathbf{z}} + \mathbf{\Phi}_{\mathbf{z}}^{\mathsf{T}}\alpha\mathbf{\Phi} + \mathbf{\Phi}_{\mathbf{z}}^{\mathsf{T}}\boldsymbol{\lambda}^{*} = \mathbf{Q} \\ &\boldsymbol{\lambda}_{i+1}^{*} = \boldsymbol{\lambda}_{i}^{*} + \alpha\mathbf{\Phi} \quad i = 1, 2, \ldots \end{aligned}}$$

- Newmark integrator

$$\boxed{\begin{aligned} \dot{\mathbf{z}}_{n+1} &= f\left(\mathbf{z}_{n+1}, \mathbf{z}_{n}, \dot{\mathbf{z}}_{n}, \ddot{\mathbf{z}}_{n}\right) \\ \ddot{\mathbf{z}}_{n+1} &= f\left(\mathbf{z}_{n+1}, \mathbf{z}_{n}, \dot{\mathbf{z}}_{n}, \ddot{\mathbf{z}}_{n}\right) \end{aligned}}$$

- Combination of formulation and integrator: Newton–Raphson

$$\mathbf{f}_{\mathbf{z}} \approx \mathbf{M} + \gamma h\mathbf{C} + \beta h^{2}\left(\mathbf{\Phi}_{\mathbf{z}}^{\mathsf{T}}\alpha\mathbf{\Phi}_{\mathbf{z}} + \mathbf{K}\right)$$

$$\mathbf{f} = \beta h^{2}\left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{z}}^{\mathsf{T}}\alpha\mathbf{\Phi} + \mathbf{\Phi}_{\mathbf{z}}^{\mathsf{T}}\boldsymbol{\lambda}^{*} - \mathbf{Q}\right)$$

- Velocity and acceleration projections

$$\mathbf{f}_{\mathbf{z}}\dot{\mathbf{z}} = \mathbf{W}\dot{\mathbf{z}}^{*} - \beta h^{2}\mathbf{\Phi}_{\mathbf{z}}^{\mathsf{T}}\alpha\mathbf{\Phi}_{t}$$

$$\mathbf{f}_{\mathbf{z}}\ddot{\mathbf{z}} = \mathbf{W}\ddot{\mathbf{z}}^{*} - \beta h^{2}\mathbf{\Phi}_{\mathbf{z}}^{\mathsf{T}}\alpha\left(\dot{\mathbf{\Phi}}_{\mathbf{z}}\dot{\mathbf{z}} + \dot{\mathbf{\Phi}}_{t}\right)$$

First Example: 2D Double Four–Bar Mechanism

- Five 1 Kg, 1 m long steel bars
- All bars can be flexible or not
- 2 static modes and 2 dynamic modes per bar
- 1 m/s initial velocity, gravity
- Integration: 5 s (2,5 turns)

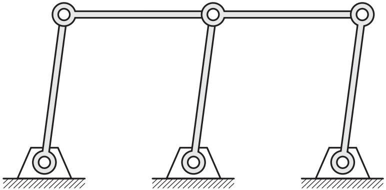


Table: Number of coordinates

# flexible bars	0	1	2	3	4	5
Absolute	6	13	20	27	34	41
Relative	5	8	11	14	17	20

First Example: 2D Double Four–Bar Mechanism

- Five 1 Kg, 1 m long steel bars
- All bars can be flexible or not
- 2 static modes and 2 dynamic modes per bar
- 1 m/s initial velocity, gravity
- Integration: 5 s (2,5 turns)

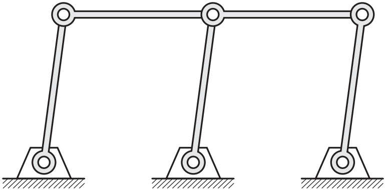
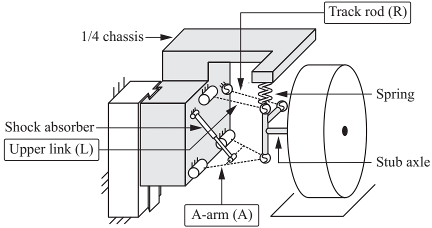


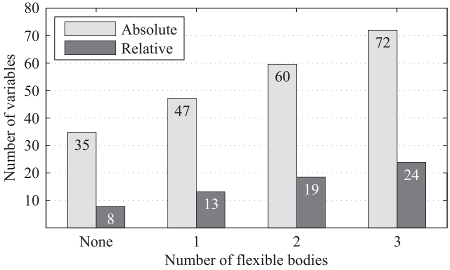
Table: CPU–times

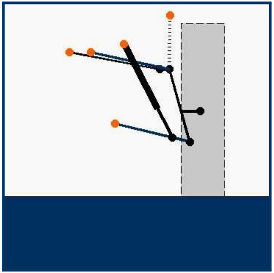
# flexible bars	0	1	2	3	4	5
Absolute	0.91	3.30	6.24	9.61	11.51	15.22
Relative	4.85	9.11	12.62	15.74	17.74	20.92

Second Example: Iltis Suspension

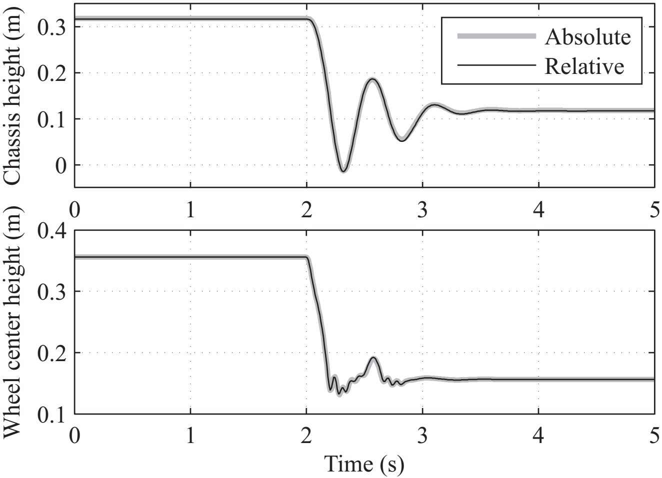
- Up to three flexible bodies
- Structural damping added
- Initial equilibrium position
- Runs down 20 cm step
- Motion integrated along 5 s
- Implemented in FORTRAN





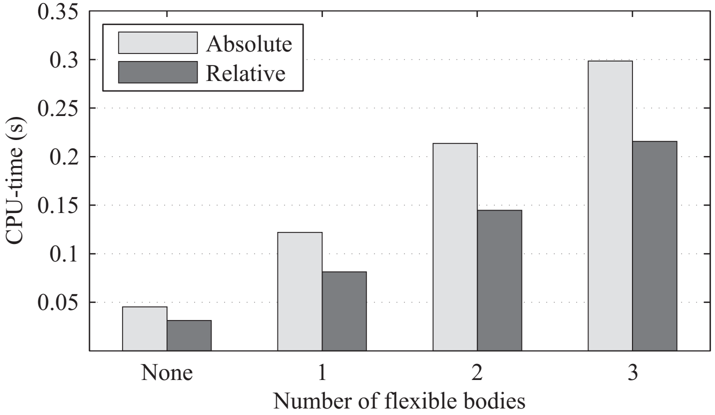


Time histories in the vertical direction



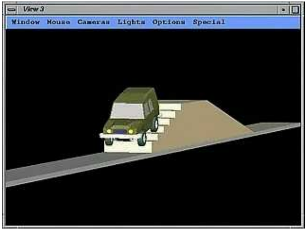
Results: Iltis Suspension

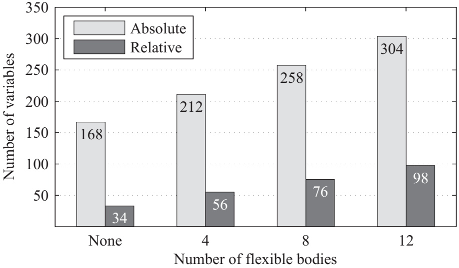
CPU–time vs. number of flexible bodies



Third Example: Full Vehicle

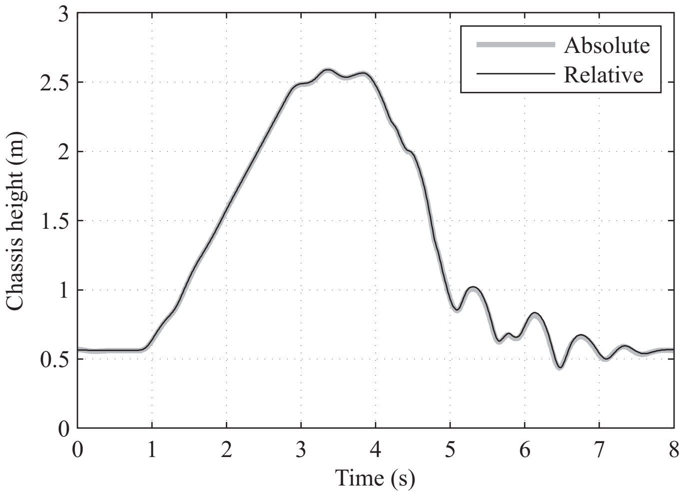
- Iltis vehicle: 4 suspensions
- Structural damping added
- Initial velocity: 5 m/s
- Road profile: ramp + steps
- Motion integrated along 8 s
- Implemented in FORTRAN





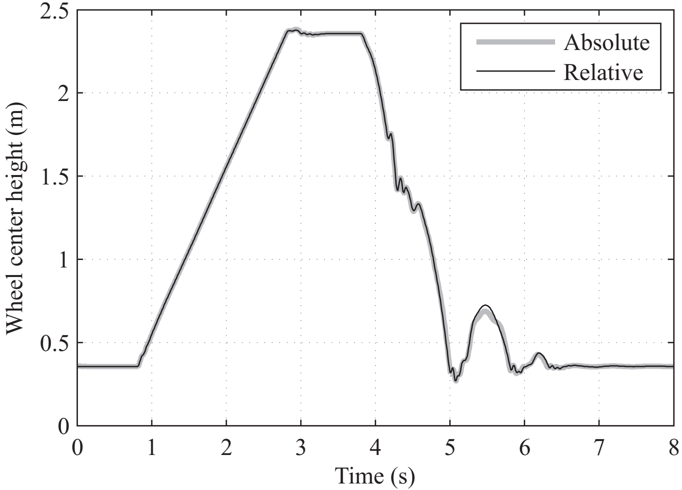
Results: Full Vehicle

Time history in vertical direction: chassis



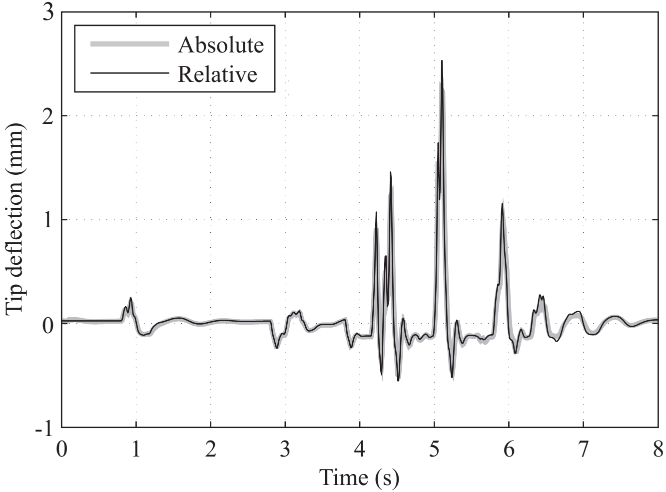
Results: Full Vehicle

Time history in vertical direction: center of front left wheel



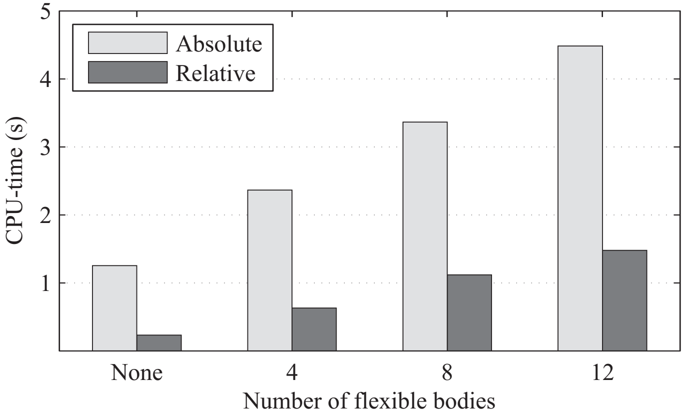
Results: Full Vehicle

Time history of the deflection of the A–arm



Results: Full Vehicle

CPU–time vs. number of flexible bodies



Conclusions of the Second Chapter

- New semi–recursive $O\left(n^3\right)$ FFR formulation successfully implemented and tested

- Very good correlation of results between both formulations

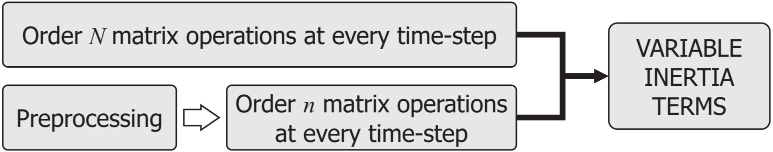
- Results in the flexible case similar to the rigid case
  - Absolute coordinates are faster in small systems
  - Relative coordinates are more efficient for large systems

- Common bottleneck: the inertia terms
  - Projection of finite element mass matrix is time–consuming
  - Solution addressed in the third chapter: Inertia Shape Integrals

Outline

Background

- Starting point: two efficient FFR formulations
  - Method in absolute (natural) coordinates
  - Method in relative coordinates
- Finite element model reduced from order N to order n
- Variable inertia terms obtained by order N velocity projections
- Bottleneck: projections take up to 80% CPU-time



- Solution: preprocessing (inertia shape integrals)
- Order n matrix operations at every time-step

Preprocessing Approach

- The mass matrix can be directly obtained by integrating $\mathbf{B}^\mathsf{T}\mathbf{B}$

$$T = \frac{1}{2}\int_V \dot{\mathbf{r}}^\mathsf{T}\dot{\mathbf{r}}\,dm = \frac{1}{2}\dot{\mathbf{q}}^\mathsf{T}\left(\int_V \mathbf{B}^\mathsf{T}\mathbf{B}\,dm\right)\dot{\mathbf{q}} \Longrightarrow \boxed{\mathbf{M} = \int_V \mathbf{B}^\mathsf{T}\mathbf{B}\,dm}$$

- Different integrals are needed depending on the formulation
  - Absolute: $\mathbf{M} = \int_V \begin{bmatrix} \mathbf{I}_3 & \bar{r}_1\mathbf{I}_3 & \bar{r}_2\mathbf{I}_3 & \bar{r}_3\mathbf{I}_3 & \mathbf{AX} \\ & \bar{r}_1^2\mathbf{I}_3 & \bar{r}_1\bar{r}_2\mathbf{I}_3 & \bar{r}_1\bar{r}_3\mathbf{I}_3 & \bar{r}_1\mathbf{AX} \\ & & \bar{r}_2^2\mathbf{I}_3 & \bar{r}_2\bar{r}_3\mathbf{I}_3 & \bar{r}_2\mathbf{AX} \\ & sym. & & \bar{r}_3^2\mathbf{I}_3 & \bar{r}_3\mathbf{AX} \\ & & & & \mathbf{X}^\mathsf{T}\mathbf{X} \end{bmatrix} dm$
  - Relative: $\bar{\mathbf{M}} = \int_V \begin{bmatrix} \mathbf{I}_3 & -\tilde{\mathbf{r}} & \mathbf{AX} \\ & -\tilde{\mathbf{r}}\tilde{\mathbf{r}} & \tilde{\mathbf{r}}\mathbf{AX} \\ sym. & & \mathbf{X}^\mathsf{T}\mathbf{X} \end{bmatrix} dm$
- Centrifugal and Coriolis forces are obtained as $-\int_V \mathbf{B}^\mathsf{T}\dot{\mathbf{B}}\dot{\mathbf{q}}\,dm$

Inertia Shape Integrals

- 13 constant integrals, including scalars, vectors and matrices
  - Mass, undeformed static moment and planar inertia tensor

$$m = \int_V dm; \quad \bar{\mathbf{m}}_u = \int_V \bar{\mathbf{r}}_u \, dm; \quad \bar{\mathbf{P}}_u = \int_V \bar{\mathbf{r}}_u \bar{\mathbf{r}}_u^\top \, dm$$

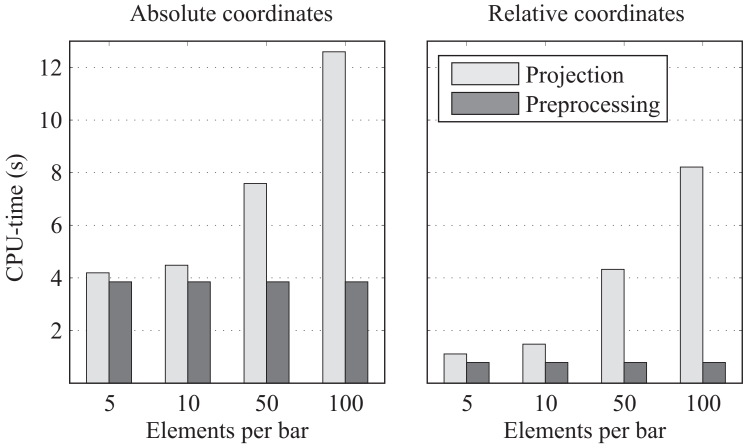
  - Four $3 \times n$ matrices

$$\mathbf{S} = \int_V \mathbf{X} \, dm; \quad \mathbf{S}^i = \int_V \bar{r}_{ui} \mathbf{X} \, dm; \quad i = 1, 2, 3$$

  - Six $n \times n$ matrices

$$\mathbf{S}^{ij} = \int_V \mathbf{X}_i^\top \mathbf{X}_j \, dm; \quad i, j = 1, 2, 3$$

CPU–time vs. Finite Element Mesh Size



Conclusions of the Third Chapter

- Preprocessing using inertia shape integrals has been implemented in both the absolute and the relative formulations

- The formulation in relative coordinates keeps its advantage over the absolute one for large size problems

- The use of preprocessing always improves efficiency
  - Improvement obtained even for small finite element models
  - Preprocessing time has no significant impact
  - The $\mathbf{B}$ matrix method is easier to implement

- Small models (<10 finite elements): $\mathbf{B}$ matrix
- Large models (>20 finite elements): inertia shape integrals

Outline

Background

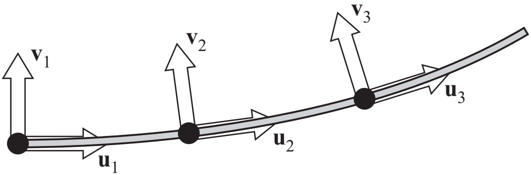
- Geometric stiffening appears in rotating beams, such as helicopter or turbine blades, increasing bending stiffness with rotation speed
- When FFR formulations are used, this effect can be lost if linear elastic displacements are assumed in the FE model

Objectives

- Objective: extend the range of usability of FFR formulations by including geometric stiffening
- Three different techniques are studied
  - Substructuring
  - Nonlinear stiffness matrix
  - Foreshortening
- These techniques are implemented and compared in absolute and relative coordinates (substructuring only in relative coordinates)

Substructuring

- The beam is divided into several substructures:



- Each substructure is a standard FFR flexible body
- Substructures are interconnected by *bracket joints*
- Most general approach, the FFR formulation is not modified
- Tested only in relative coordinates
  - Natural coordinates: $12 + n_m$ variables per substructure
  - Relative coordinates: n_m variables per substructure

Nonlinear Stiffness Matrix: Potential Energy

- Strain energy of an Euler–Bernoulli beam

$$U = \underbrace{\frac{1}{2}\int_0^L EA{u'_0}^2\,dx + \frac{1}{2}\int_0^L EI{v''_0}^2\,dx}_{Linear\ formulation}$$
$$+ \underbrace{\frac{1}{2}\int_0^L EAu'_0{v'_0}^2\,dx}_{First\ nonlinear} + \underbrace{\frac{1}{8}\int_0^L EA{v'_0}^4\,dx}_{Second\ nonlinear}$$

- Linear formulation: only the first two terms are retained
  - Axial and transversal displacements are independent
- First nonlinear formulation: the third term is added
  - Introduces coupling between axial and transversal displacement
- Second nonlinear formulation: full strain energy expression

Nonlinear Stiffness Matrix: Elastic Forces

- Linear formulation
  - Constant stiffness matrix
  - No coupling between axial and transversal stiffness

$$\mathbf{F}_{el} = -\mathbf{K}_L \mathbf{y}$$

- First nonlinear formulation
  - Variable stiffness matrix
  - $\mathbf{K}_G$ couples axial and transversal stiffness

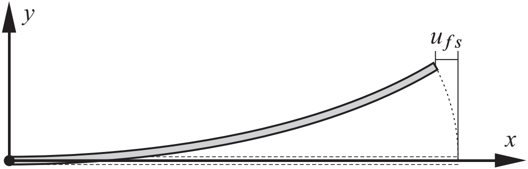
$$\mathbf{F}_{el} = -\left(\mathbf{K}_L + \mathbf{K}_G\right)\mathbf{y}; \quad \mathbf{K}_G = \sum_{i=1}^{ns} \eta_i \mathbf{K}_{Gi} + \sum_{j=1}^{nd} \xi_j \mathbf{K}_{Gj}$$

- Second nonlinear formulation
  - Highly nonlinear stiffness matrix

$$\mathbf{F}_{el} = -\left(\mathbf{K}_L + \mathbf{K}_G + \mathbf{K}_H\right)\mathbf{y} + \mathbf{Q}_g$$

Foreshortening

- Foreshortening: axial shortening produced by deflection



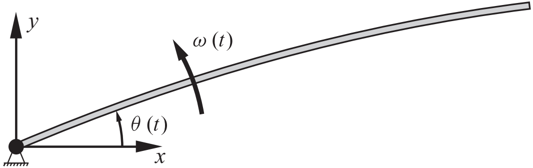
- Modified axial displacement

$$u_0 = s + u_{fs}; \quad u_{fs}(x) = -\frac{1}{2}\int_{x_0}^{x} {v_0'}^2\, dx$$

- Strain energy equivalent to second nonlinear formulation
- Introduced in the axial components of the mode shapes $\mathbf{X}$
  - It renders the $\mathbf{X}$ matrix variable $\Longrightarrow$ $\mathbf{B}^*$ projection
  - Linear elastic forces with unmodified $\mathbf{K}_L$ matrix
  - Geometric stiffening is introduced at the kinematics level
- Captures the effect with no axial modes

System Under Test: Rotating Beam

- Steel beam pinned at one end



- Guided rotation about the origin

$$\omega(t) = \begin{cases} \dfrac{\Omega_s}{T_s}\left[t - \left(\dfrac{T_s}{2\pi}\right)\sin\left(\dfrac{2\pi t}{T_s}\right)\right] & 0 \leq t < T_s \\ \Omega_s & T_s \leq t \end{cases}$$

- 2D and 3D cases studied; motion is integrated along 20 s
- Deflection at the tip is measured for Ω_s=6 rad/s, T_s=15 s
- Results are compared to a reference solution (ANCF or FEM)

Results: Horizontal Deflection in the 2D Case

Linear formulation

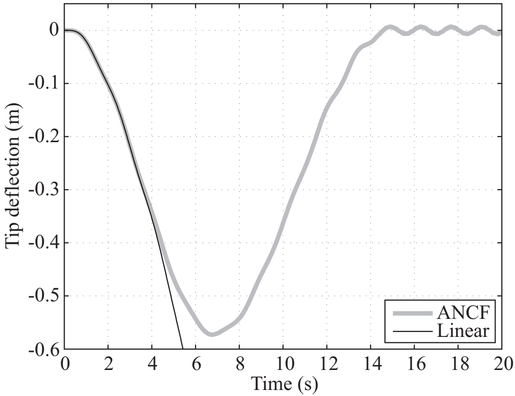


Table: CPU–times (s)

Method	AC[1]	RC[2]
FNL1	0.266	0.094
FNL2	0.297	0.125
FS0	**0.266**	**0.094**

[1]AC: Absolute Coordinates
[2]RC: Relative Coordinates

First nonlinear formulation

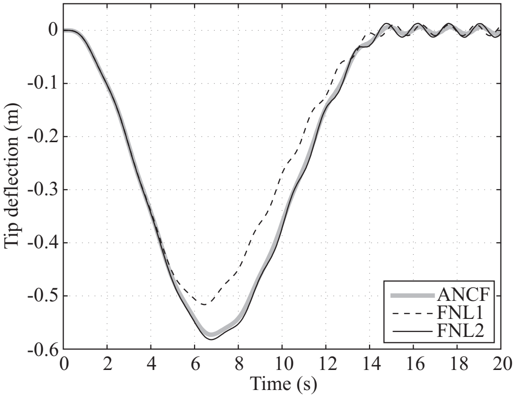


Table: CPU–times (s)

Method	AC[1]	RC[2]
FNL1	0.266	0.094
FNL2	0.297	0.125
FS0	**0.266**	**0.094**

[1]AC: Absolute Coordinates
[2]RC: Relative Coordinates

Results: Horizontal Deflection in the 2D Case

Foreshortening

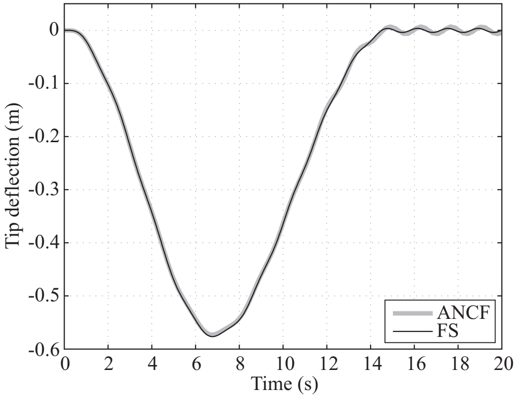


Table: CPU–times (s)

Method	AC[1]	RC[2]
FNL1	0.266	0.094
FNL2	0.297	0.125
FS0	**0.266**	**0.094**

[1]AC: Absolute Coordinates
[2]RC: Relative Coordinates

Results: Horizontal Deflection in the 2D Case

Foreshortening

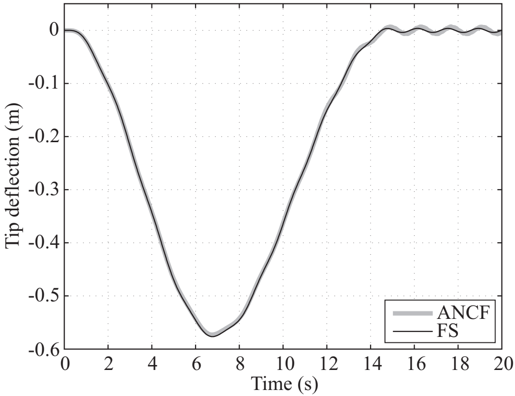
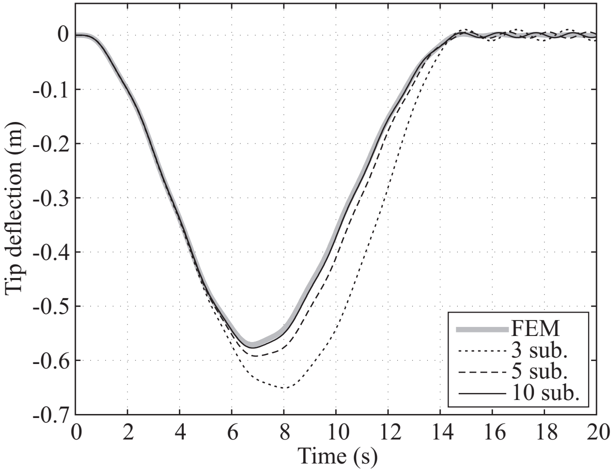


Table: CPU–times (s)

Method	AC[1]	RC[2]
FNL1	0.266	0.094
FNL2	0.297	0.125
FS0	**0.266**	**0.094**

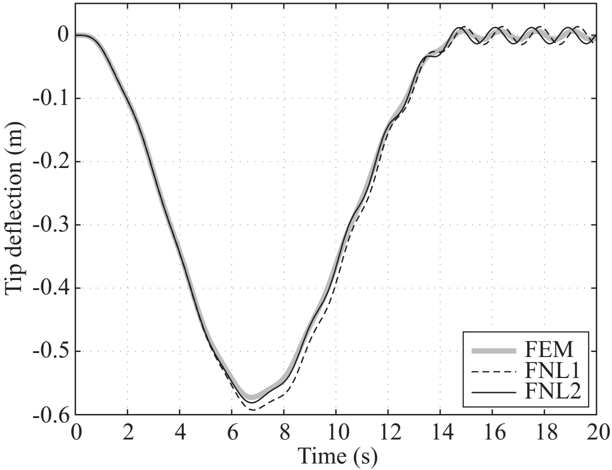
[1]AC: Absolute Coordinates
[2]RC: Relative Coordinates

Substructuring



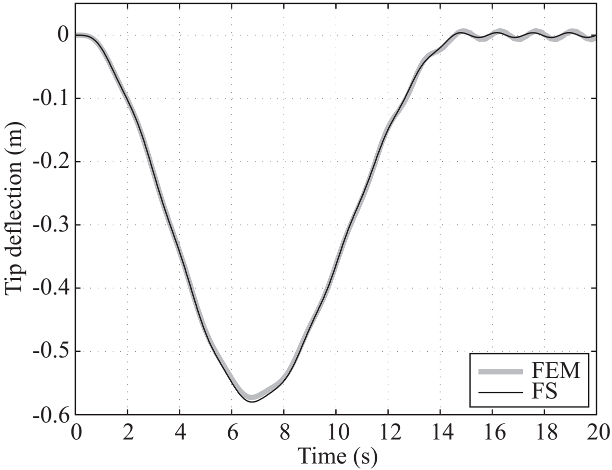
Results: Horizontal Deflection in the 3D Case

First nonlinear formulation



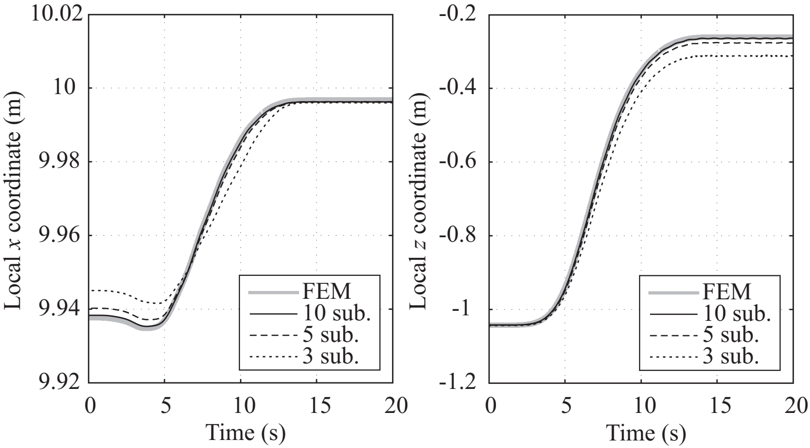
Results: Horizontal Deflection in the 3D Case

Foreshortening



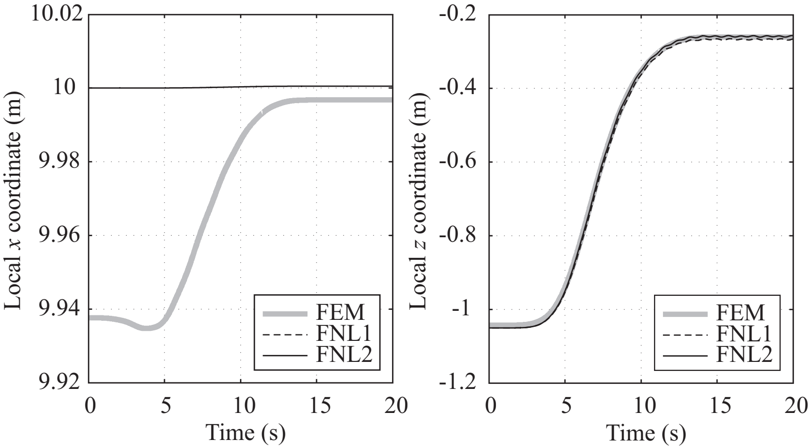
Results: Vertical and Axial Displacements

Substructuring



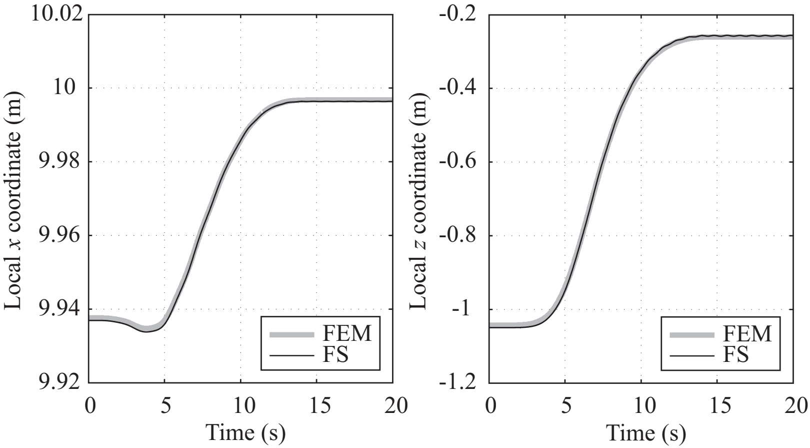
Results: Vertical and Axial Displacements

First nonlinear formulation



Results: Vertical and Axial Displacements

Foreshortening



Results: 3D Efficiency and Accuracy

3D spin–up beam results

Method		SB10	SB20	FNL1	FNL2	FS0
CPU–time (s)	AC[1]	–	–	0.271	0.286	**0.250**
	RC[2]	1.312	4.578	0.105	0.125	**0.105**
Δx (mm)		0.502	**0.320**	27.944	27.947	0.500
Δy (mm)		3.193	**1.481**	10.680	5.757	3.140
Δz (mm)		2.329	**0.742**	7.606	3.876	4.796

[1]AC: Absolute Coordinates [2]RC: Relative Coordinates

Conclusions of the Fourth Chapter

- Linear FFR method cannot capture geometric stiffening effect
- Substructuring method
  - Best accuracy, increasing with the number of substructures
  - Easy implementation into FFR codes, no modifications required
  - High CPU–times if compared to other methods
- First nonlinear formulation
  - Very fast and easy to implement, only affects the $\mathbf{K}$ matrix
  - No foreshortening $\implies$ highest error in axial direction
  - Axial modes are required
- Foreshortening
  - Almost as accurate and much faster than substructuring
  - No axial modes are required $\implies$ numerical integrator friendly
  - More involved implementation, requires preprocessing

Outline

Final Conclusions

- New semi–recursive $O\left(n^3\right)$ FFR formulation
  - Efficiency improvement for systems above 25 variables
  - More robust than the formulation in absolute coordinates
  - More involved implementation
- Shape integrals preprocessing implemented in both formulations
  - Always more efficient than $\mathbf{B}^*$ matrix projection
  - Projection is much simpler and fast enough for small models
  - It is also more convenient for including foreshortening
- Three methods for modeling nonlinear beams have been tested
  - Substructuring is the most accurate approach
  - The $\mathbf{K}_G$ method is extremely simple and yields acceptable results
  - Foreshortening obtains the best efficiency/accuracy ratio

Future Research

- Study of different model reduction methods
  - Krylov subspaces are based on response characteristics
- Mode selection techniques
  - The selection of mode shapes is left to the analyst
  - Development of automated techniques
- Further optimization of the calculation of the inertia terms
  - Check relative weights of the different terms
  - The number of operations depends on the reference conditions