

**MBDyn Theory
and Developer's Manual
Version develop**

Pierangelo Masarati

DIPARTIMENTO DI INGEGNERIA AEROSPAZIALE
POLITECNICO DI MILANO

August 26, 2022

Contents

1	Introduction	7
2	Parsing	8
2.1	HighParser	8
2.1.1	Traditional Usage	8
2.1.2	Table-Driven Usage	9
2.2	LowParser	11
2.3	MathParser	11
3	Solvers	12
3.1	Matrix classes	12
3.2	Sparse matrices	12
3.3	Linear solvers	12
3.4	Non linear solvers	12
3.5	Parallel solver	12
3.5.1	Partitioning	12
3.6	Convergence check	13
4	Orientation Handling	15
4.1	Euler Angles	15
4.2	Orientation Vector	16
5	Integration	18
5.1	Nodal rotation	18
5.2	Integrators	18
6	Solution Phases	19
6.1	Initial Assembly	19
6.2	Initial Value Problem	22
6.2.1	Initial Derivatives	22
6.2.2	Dummy Steps	23
6.2.3	Regular Steps	23
6.3	Inverse Dynamics Problem	23
6.3.1	Nomenclature	24
6.3.2	Fully Actuated, Collocated Problem	24
6.3.3	Fully Actuated, Non-Collocated Problem	25
6.3.4	Underdetermined, Underactuated but Collocated Problem	25
6.3.5	Underdetermined, Overcontrolled Problem	27

7 Data Structure	30
7.1 Constitutive Laws	30
7.2 ExpandableRowVector	30
8 Nodes	33
8.1 Structural Nodes	33
8.1.1 Dynamic Structural Nodes	33
8.1.2 Static Structural Nodes	36
8.1.3 Dummy Nodes	37
8.1.4 Relative Motion	37
8.1.5 Motion Expressed in a Relative Reference Frame	39
8.1.6 Dynamics in a Relative Reference Frame	40
8.1.7 Airstream Velocity	41
8.1.8 Implementation Notes	42
8.1.9 Pseudo-Velocities Approach	43
9 Constraints	47
9.1 Algebraic Constraints	47
9.1.1 Clamp Joint	48
9.1.2 Distance Joint	49
9.1.3 Distance Joint With Offsets	50
9.1.4 Spherical hinge	51
9.1.5 Revolute hinge	52
9.1.6 Inline	55
9.1.7 Drive Hinge	57
9.1.8 Drive Displacement	58
9.1.9 Drive Displacement Pin	60
9.1.10 Imposed Displacement	61
9.1.11 Imposed Displacement Pin	62
9.1.12 Total Joint	63
9.1.13 Total Pin Joint	69
9.1.14 Gimbal	72
9.1.15 Screw Joint	74
9.1.16 Strapdown Sensor	79
9.2 Deformable Constraints	80
9.2.1 Rod With Offsets	80
9.2.2 Deformable Hinge	83
9.2.3 Deformable Displacement Joint	93
9.2.4 Deformable Joint	99
9.2.5 Deformable Axial Joint	105
9.3 Viscous Body	106
9.4 Modal Element	107
9.4.1 Kinematics	107
9.4.2 Physics: Orthogonality	110
9.4.3 Simplifications	111
9.4.4 Invariants	112
9.4.5 Interfacing	114

10 Beam Element	116
10.1 Generalized strains, strain rates and their linearization	116
10.2 Fully coupled piezoelectric beam	120
11 Shell Element	121
11.1 Variational principle	121
11.1.1 Strain Rate	122
11.2 Finite element discretization and notation	123
11.3 Linearization	124
11.4 Structural Damping	125
11.5 Implementation	125
11.5.1 Orientation interpolation	125
11.5.2 Position interpolation	126
11.5.3 Enhancing strains interpolation	127
11.5.4 Compatible strains interpolation	128
11.5.5 ANS	129
11.5.6 Forces	131
11.5.7 Jacobian matrix	131
11.5.8 Residual	132
12 Aerodynamic Elements	134
12.1 Linearization of 2D Aerodynamic Forces and Moments	134
12.2 Numerical Linearization of Sectional Forces	136
12.3 Aerodynamic Forces with Internal States	136
12.4 Aerodynamic body	137
12.5 Aerodynamic Beam (3 Nodes)	138
12.6 Aerodynamic Beam (2 Nodes)	141
12.7 Unsteady aerodynamics model	145
12.7.1 Perturbation of the Equations	147
12.7.2 Perturbation of the aerodynamic forces	150
12.7.3 Perturbation of the aerodynamic moments	151
12.7.4 Finite difference version	152
12.7.5 Perturbation of the equations	153
12.7.6 Perturbation of the aerodynamic forces	155
12.7.7 Perturbation of the aerodynamic moments	156
12.8 Rotor	156
12.8.1 Uniform Inflow Model	158
12.8.2 Glauert Model	158
12.8.3 Mangler-Squire Model	159
12.8.4 Dynamic Inflow Model	160
12.9 Aero Modal	162
12.9.1 Clamped	162
12.9.2 Free	163
13 Forces	164
13.1 Abstract Force	164
13.1.1 Abstract	164
13.1.2 Abstract Internal	164
13.2 Structural Forces	164
13.2.1 Force	164

13.2.2	Couple	164
13.3	Modal	164
13.4	External	164
13.4.1	External Structural	165
13.4.2	External Structural Mapping	166
13.4.3	External Modal	167
13.4.4	External Modal Mapping	167
13.4.5	Client Library	167
14	Hydraulic Library	169
14.1	Hydraulic Fluids	169
14.2	Hydraulic Nodes	169
14.3	Hydraulic Elements	169
14.3.1	Accumulator	169
14.3.2	Actuator	169
14.3.3	Dynamic Pipe	170
14.3.4	Pressure Flow Control Valve	173
A	On the optimization of n-sub-step composite time integration methods	175
B	Performance of implicit A-stable time integration methods for multibody system dynamics	200

List of Figures

9.1	3D screw thread sketch.	77
9.2	2D screw thread sketch	78
11.1	Shell finite element	123

List of Tables

12.1 Coefficients of the Wagner indicial response approximation of Theodorsen's function ([1, 2])	145
12.2 Glauert inflow model (source: Leishman [2])	159
12.3 Glauert inflow model as implemented in MBDyn	159

Chapter 1

Introduction

This document describes the formulation MBDyn, the free general-purpose multibody dynamics software, relies on. It also describes implementation-related aspects.

The document is far from complete; in fact, its preparation started at a late stage of the project, when many parts of the software were already completed and in use. From that time on, many modifications and improvements have been steadily documented and, occasionally, the theory manual anticipated the development (as should always happen, I know).

The Developers committed themselves to keeping it at least up to date with new developments; undocumented stuff should be documented as soon as it needs modifications or refactoring.

Alessandro Fumagalli, Marco Morandini and Mattia Mattaboni contributed significantly to this document.

Chapter 2

Parsing

In MBDyn there are different levels of parsing. Input file parsing suffers from a scattered and occasionally outdated design. The need to preserve backwards compatibility with existing models restrained so far from entirely redesign it, although selected improvements occur over time.

MBDyn provides support for the implementation of new functionalities, and to parse their input. Parsing is delegated to a dedicated object, the `MBDynParser`, which inherits from the `HighParser`, a higher-level parsing object with bits of MBDyn's syntax built-in in a somewhat modular way. It exploits the functionalities of the `LowParser`, a lower-level parsing object that deals with tokenizing an input stream based on the expected tokens. Whenever appropriate (e.g. whenever a number is expected), control is delegated to the `MathParser`, which allows to parse and evaluate sequences of mathematical expressions, including variable declaration and definition. Variables are saved in a table as soon as they appear, and can be used by subsequent expressions when the mathematical parser is called again.

The traditional approach to data parsing consists in defining a table of keywords that are looked up and, based on the corresponding key code, by executing the appropriate code in a switch-case block.

The code is being gradually moved to a newer approach based on sets of associative arrays that map keywords to the functional objects that are used to parse the related items.

Although no significant improvement results in parsing of existing data types, this approach allows to register new data types run-time, e.g. from a run-time loaded module, thus easing the extension and the customization of the code.

2.1 HighParser

2.1.1 Traditional Usage

The `HighParser` class and its descendants use a `KeyTable` object containing a list of legal keywords to return a valid keyword index when `HighParser::GetWord()`, and significantly `HighParser::GetDescription()` are invoked. The `KeyTable` can be changed during parsing. `KeyTable` is a class. Its constructor takes a pointer to an array of strings and a reference to the `HighParser` object. The last string in the array must be null. The `KeyTable` class constructor keeps track of previous `KeyTable` objects in the `HighParser`, and restores them upon destruction.

The suggested usage inside a stacked call sequence of parsing functions is

```
Part *
read_part(HighParser& HP)
{
    // prepare names
```

```

enum KeyWord { KEYWORD1, KEYWORD2, KEYWORD_LAST };
char *key_table_array[] = { "keyword1", "keyword2", 0 };
// build KeyTable class
KeyTable k(HP, key_table_array);
Part *returned_object = 0;
// parse input
do {
    switch (KeyWord(HP.GetWord())) {
    default:
        // do something...
        break;
    case KEYWORD1:
        // ...and build returned_object
        return returned_object;
    }
} while (true);
}

void
read_all(HighParser& HP)
{
    // prepare names
enum KeyWord { KEY1, KEY2, PART, KEY_LAST };
char *keytable[] = { "key1", "key2", "part", 0 };
// build KeyTable class
KeyTable k(HP, keytable);
// do something
Part *part = 0;
do {
    switch (KeyWord(HP.GetWord())) {
    default:
        // do something...
        break;
    case PART:
        // read part
        Part *part = read_part(HP);
        break;
    }
} while (true);
}

```

Here the KeyTable set by function `read_all()` is automatically restored after the call to `read_part()`; `read_part()` temporarily changes the KeyTable used by the parser.

2.1.2 Table-Driven Usage

The table-driven usage is based on defining a functional object that is able to parse a data type:

```

class Datum {
public:

```

```

    virtual ~Datum(void) {};
};

struct DatumRead {
    virtual ~DatumRead(void) {};
    virtual Datum *Read(MBDynParser &HP) const = 0;
};

```

and a container for its descendants:

```

typedef std::string KeyType;
typedef std::map<KeyType, DatumRead *> DatumMapType;
DatumMapType DatumMap;

```

Then, a functional object for each specific datum type is derived from `DatumRead` ...

```

class MyDatum : public Datum {
    // ...
public:
    Datum(int);
    // ...
}

struct MyDatumRead : public DatumRead {
    Datum *Read(MBDynParser &HP) const {
        return new MyDatum(HP.GetInt());
    };
};

```

...and stored exactly once into the associative container by means of a dedicated helper:

```

bool
SetDatumRead(KeyType key, DatumRead *rf)
{
    return DatumMap.insert(DatumMapType::value_type(key, rf)).second;
}

// somewhere early in the code ...
SetDatumRead("mydatum", new MyDatumRead);

// note: somewhere else later, in the code, place
for (DatumMapType::iterator i = DatumMap.begin();
     i != DatumMap.end();
     i++)
{
    delete i->second;
}
DatumMap.clear();

```

The parsing function is something like

```

Datum *
ReadDatum(MBDynParser &HP)

```

```

{
    KeyType key(HP.GetStringWithDelims());
    if (key.c_str() == 0) {
        // error ...
        return 0;
    }
    DatumMapType::iterator i = DatumMap.find(key);
    if (i == DatumMap.end()) {
        // error ...
        return 0;
    }
    return i->second->Read(HP);
}

```

Only the `MyDatum` portions need be added for each new datum type; they can be declared, defined and registered anywhere in the code, including in run-time loaded modules.

Currently, drives, constitutive laws and scalar functions are handled according to this scheme; examples are provided in `modules/module-drive/`, `modules/module-constlaw/` and `modules/module-scalarfunc/`. More types will be reworked accordingly.

2.2 LowParser

...

2.3 MathParser

...

Chapter 3

Solvers

...

3.1 Matrix classes

3.2 Sparse matrices

3.3 Linear solvers

LinearSolver classes wraps linear solvers

SolutionManagers classes deals with the solution of linear systems. They own a pointer to a LinearSolver, where they allocate the underlying linear solver. SolutionManager::MatrInitialize() is called when the structure of the underlying spares matrix changes. SolutionManager::MatrReset(), is called to delete a factorization. It usually calls LinearSolver::Reset(). The matrix has to be explicitly zeroed before a Jacobian matrix assembly.

3.4 Non linear solvers

...

3.5 Parallel solver

3.5.1 Partitioning

`iTotVertices` is equal to the sum of nodes and elements. It is made in this way because we want the partitioner to generate a twofold subdivision:

- a subdivision related to elements; this subdivision is done in order to share the computational load during the assembly phase;
- a subdivision related to nodes, which is necessary for the solving phase with the substructuring method.

Of course this two partitions must be connected, so we build created a graph which is made of nodes and elements as vertex. The connection between vertexes are only between nodes and elements. There is no node to node or element to element connection.

`pVertexWgts` contains what we call the computational weight of each entity, so nodes have weight null while elements has a weight related to the dimension of the submatrix of the Jacobian matrix assembled by each one of them.

`pCommWgts` contains the communication weights (see metis documentation) which are a measure of the quantity of data which needs to be sent if the i-th vertex is part of an interface between different partitions. This means that nodes have a `CommWgts` equal to the number of DOFs, while elements have a weight equal to any internal DOFs they possess.

3.6 Convergence check

MDByn needs to solve a set of nonlinear equations

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}, \quad (3.1)$$

where \mathbf{f} is the so-called residual vector. The standard `norm` convergence check is based on the norm

$$f = \sqrt{\mathbf{f}^T \mathbf{f}} \quad (3.2)$$

of vector \mathbf{f} . While this often works in practice it has two shortcoming:

1. one doesn't know the generalized forces magnitude of the model, that is, a residual norm equal to 1 can be perfectly fine if the order of magnitude of the generalized forces transmitted by the structure is of 1E6, while is not ok if the order of magnitude is 10;
2. different components of vector \mathbf{f} do have different physical dimensions, and it makes no sense to sum them together.

The `relnorm` test is designed to solve the first shorcoming. In addition to assembling the residual component f_i as

$$f_i = \sum_{e=1}^{\text{elements of the model}} f_{i(e)}(\mathbf{x}), \quad (3.3)$$

also the absolute values of the different contribution from the elements are assembled into a new vector $\tilde{\mathbf{f}}$ with components

$$\tilde{f}_i = \sum_{e=1}^{\text{elements of the model}} |f_{i(e)}(\mathbf{x})|. \quad (3.4)$$

The convergence test is then performed over

$$f = \frac{\sqrt{\mathbf{f}^T \mathbf{f}}}{\sqrt{\tilde{\mathbf{f}}^T \tilde{\mathbf{f}}}} \quad (3.5)$$

so that the residual is automatically scaled with respect to the generalized force magnitude appearing in the model.

The `sepnorm` test tries to solve both issues. Beside assembling the additional vector $\tilde{\mathbf{f}}$, just like the `relnorm` test, it partitions the equations into N independent set $\mathbf{f}_{[j]}$ and $\tilde{\mathbf{f}}_{[j]}$, with $j \in [1, N]$, one foreach

physical dimension apperaring in the residual vector, so that every components $\mathbf{f}_{i[j]}$ of the subvector $\mathbf{f}_{[j]}$ has the same physical dimension. The relative residual test is then computed independently for each set

$$f_{[j]} = \frac{\sqrt{\mathbf{f}_{[j]}^T \mathbf{f}_{[j]}}}{\sqrt{\tilde{\mathbf{f}}_{[j]}^T \tilde{\mathbf{f}}_{[j]}}} \quad (3.6)$$

and the test returns the worst $f_{[j]}, \max_j(f_{[j]})$.

In order to identify the set of equations that are inactive (for which either $\sqrt{\tilde{\mathbf{f}}_{[j]}^T \tilde{\mathbf{f}}_{[j]}} = 0$ or $f_{[j]} \approx 1$ because the norms are very small and comes mostly from numerical errors),

1. $f_{[j]} = 0$ if $\sqrt{\tilde{\mathbf{f}}_{[j]}^T \tilde{\mathbf{f}}_{[j]}} = 0$
AND
2. $f_{[j]} = 0$ if $\sqrt{\mathbf{f}_{[j]}^T \mathbf{f}_{[j]}} / \sqrt{\tilde{\mathbf{f}}_{[j]}^T \tilde{\mathbf{f}}_{[j]}} > \epsilon_1$ and $\sqrt{\tilde{\mathbf{f}}_{[j]}^T \tilde{\mathbf{f}}_{[j]}} < \epsilon_2 * \max_j(\sqrt{\tilde{\mathbf{f}}_{[j]}^T \tilde{\mathbf{f}}_{[j]}})$

where the coefficients ϵ_1 and ϵ_2 are hard coded, $\epsilon_1 = 1.E - 1$ and $\epsilon_2 = 1.E - 5$. One could consider, in the future, to allow reading them from the input file.

Chapter 4

Orientation Handling

The orientation of structural nodes is handled by storing the orientation matrix of each node, and updating it during prediction and correction by incremental orientations.

The code uses Gibbs-Rodrigues parameters to account for incremental rotations. These parameters allow very efficient implementation, since they only require few algebraic operations to obtain the orientation matrix and related entities. However, they require the absolute value of the orientation to be limited; this requirement is usually overcome by accuracy requirements, so it is not viewed as a strong limitation.

One main advantage of this approach is that only three parameters are required to handle orientations. A main drawback of this approach is that it is quite difficult to keep track of how large a change in orientation occurred to a given node, since simply converting the orientation matrix into any other representation of a finite rotation typically results in the minimal rotation that yields that matrix.

MBDyn can output rotations in the following representations:

- Euler angles according to the 123 sequence;
- the orientation vector;
- the orientation matrix.

4.1 Euler Angles

Consider an orientation expressed by matrix \mathbf{R} . It can be expressed as the result of a precise sequence of rotations about three distinct axes. The 123 sequence means that matrix \mathbf{R} is obtained as a sequence of three rotations: the first about axis 1; the second about axis 2 as it results after the first rotation; the third about axis 3 as it results after the second rotation, namely:

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}, \quad (4.1)$$

$$\mathbf{R}_2 = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}, \quad (4.2)$$

$$\mathbf{R}_3 = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.3)$$

piled up, must equal \mathbf{R} :

$$\begin{aligned} \mathbf{R} &= \mathbf{R}_1 \mathbf{R}_2 \mathbf{R}_3 \\ &= \begin{bmatrix} \cos \beta \cos \gamma & -\cos \beta \sin \gamma & \sin \beta \\ \sin \alpha \sin \beta \cos \gamma + \cos \alpha \sin \gamma & -\sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & -\sin \alpha \cos \beta \\ -\cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \cos \alpha \sin \beta \sin \gamma + \sin \alpha \cos \gamma & \cos \alpha \cos \beta \end{bmatrix}. \end{aligned} \quad (4.4)$$

The angles can then be computed by means of a simple, partially recursive algorithm:

$$\alpha = -\tan^{-1} \left(\frac{\mathbf{R}_{23}}{\mathbf{R}_{33}} \right) \quad (4.5)$$

$$\beta = \tan^{-1} \left(\frac{\mathbf{R}_{13}}{\cos \alpha \mathbf{R}_{33} - \sin \alpha \mathbf{R}_{23}} \right) \quad (4.6)$$

$$\gamma = \tan^{-1} \left(\frac{\cos \alpha \mathbf{R}_{21} + \sin \alpha \mathbf{R}_{31}}{\cos \alpha \mathbf{R}_{22} + \sin \alpha \mathbf{R}_{32}} \right) \quad (4.7)$$

The use of the function `atan2(double dSin, double dCos)` eliminates the risk of divisions by zero or of excessive loss of precision.

4.2 Orientation Vector

An orientation matrix can be represented as a finite rotation about an axis, in the form

$$\mathbf{R} = \mathbf{I} + \sin \phi \mathbf{n} \times + (1 - \cos \phi) \mathbf{n} \times \mathbf{n} \times \quad (4.8)$$

where ϕ is the amplitude of the rotation and \mathbf{n} is the unit vector that represents the rotation axis.

Simple algebra manipulation shows that the trace of the orientation matrix corresponds to

$$\text{tr}(\mathbf{R}) = \text{tr}(\mathbf{I}) + (1 - \cos \phi) \text{tr}(\mathbf{n} \times \mathbf{n} \times) \quad (4.9)$$

$$= 3 - 2(1 - \cos \phi), \quad (4.10)$$

since

$$\mathbf{n} \times \mathbf{n} \times = \mathbf{n} \mathbf{n}^T - \mathbf{I}, \quad (4.11)$$

so

$$\cos \phi = \frac{\text{tr}(\mathbf{R}) - 1}{2}. \quad (4.12)$$

At the same time, the skew-symmetric portion of \mathbf{R} is

$$\text{skw}(\mathbf{R}) = \sin \phi \mathbf{n} \times, \quad (4.13)$$

so

$$\text{ax}(\text{skw}(\mathbf{R})) = \sin \phi \ \mathbf{n} \quad (4.14)$$

which yields the amplitude of the rotation,

$$\sin \phi = \text{norm}(\text{ax}(\text{skw}(\mathbf{R}))), \quad (4.15)$$

and the direction of the rotation axis,

$$\mathbf{n} = \frac{1}{\sin \phi} \text{ax}(\text{skw}(\mathbf{R})), \quad (4.16)$$

while ϕ can be computed from its sine and cosine as

$$\phi = \tan^{-1} \left(\frac{\sin \phi}{\cos \phi} \right). \quad (4.17)$$

Chapter 5

Integration

differential variable: a variable is declared differential in `SimulationEntity::GetDofType()` by returning `DofOrder::DIFFERENTIAL`.

The increment of the value of a differential variable is equal to $\Delta x = dCoef \Delta \dot{x}$. When writing the Jacobian matrix, this must be considered; as a consequence, for an equation $f = 0$ (the residual is $-f$) the linearization is $f_{/\dot{x}} + f_{/x} * dCoef * \Delta \dot{x} = -f$, as $dCoef * \Delta \dot{x} = \Delta x$.

algebraic variable: a variable is declared algebraic in `SimulationEntity::GetDofType()` by returning `DofOrder::ALGEBRAIC`.

The increment of the value of an algebraic variable is the increment of the variable.

differential equation: an equation is declared differential in `SimulationEntity::GetEqType()` by returning `DofOrder::DIFFERENTIAL`.

An equation $f = 0$ must be declared differential if $f_{/\dot{x}}$ is not null. The residual is $-f$, and its linearization is: $(f_{/\dot{x}} + f_{/x} * dCoef) * \Delta \dot{x} = -f$.

algebraic equation: an equation is declared algebraic in `SimulationEntity::GetEqType()` by returning `DofOrder::ALGEBRAIC`.

An equation $f = 0$ can be declared algebraic iff $f_{/\dot{x}}$ is structurally null (e.g. regardless of the values the state may assume) and x is not algebraic. If an equation $f(x, t) = 0$ is declared differential, the residual is $-f$, and its linearization is: $f_{/x} * dCoef * \Delta \dot{x} = -f$. If the equation can be declared algebraic, it can be divided by `dCoef`: $f/dCoef = 0$, with residual $-f/dCoef$, and linearization $f_{/x} * \Delta \dot{x} = -f/dCoef$. This helps scaling the equations. Clearly, this has no sense if x is algebraic, or if $f_{/\dot{x}} \neq 0$.

5.1 Nodal rotation

The rotational DOF unknown during `AssRes()` and `AssJac()` are the increment of (Gibbs-Rodrigues) rotation parameters with respect to the reference configuration. Of course the increment of the parameter is $\Delta g = dCoef \Delta \dot{g}$. The increment of angular velocity is $\Delta \omega = G \Delta \dot{g} + \Delta G \dot{g} - \omega_{ref} \times G \Delta g$, where $G(g)$ is the tensor relating g_δ to δg , and ω_{ref} is the nodal reference angular velocity (Wref). We assume $G = I$ and $\Delta G = 0$, so that $\Delta \omega = \Delta \dot{g} - \omega_{ref} \times \Delta g$ and so $\Delta \omega = \Delta \dot{g} - \omega_{ref} \times \Delta \dot{g} * dCoef$.

5.2 Integrators

Most of MBDyn's integrators are documented in [3, 4], that are distributed verbatim in Appendix ??, respectively.

Chapter 6

Solution Phases

6.1 Initial Assembly

This phase only involves some of the structural elements. It is intended to ensure that the initial configuration and velocity complies with the constraint equations. It is not performed if the **control data** block contains the statement `skip initial joint assembly`.

To allow non-compliant system analysis, the initial configuration and velocity can be changed by this phase. To this purpose, the nodal positions, orientations, velocities and angular velocities are grounded by dummy springs, acting as penalty functions. The springs can be set on a node basis, and separately for configuration and velocity, to allow to selectively enforce the initial configuration.

The problem can be stated as follows:

$$\mathbf{K}\mathbf{x} + \phi_{/\mathbf{x}}^T \boldsymbol{\lambda}_\phi + \mathbf{A}^T \boldsymbol{\lambda}_{\mathbf{A}} = \mathbf{K}\mathbf{x}_0 + \mathbf{f} \quad (6.1)$$

$$\mathbf{C}\mathbf{v} + \phi_{/\mathbf{x}}^T \boldsymbol{\mu} = \mathbf{C}\mathbf{v}_0 \quad (6.2)$$

$$\phi(\mathbf{x}, t) = \mathbf{0} \quad (6.3)$$

$$\mathbf{A}(\mathbf{x}, t)\mathbf{v} + \mathbf{b}(\mathbf{x}, t) = \mathbf{0} \quad (6.4)$$

$$\phi_{/\mathbf{x}}\mathbf{v} + \phi_{/\mathbf{t}} = \mathbf{0} \quad (6.5)$$

where Equation (6.3) and (6.4) respectively contain the holonomic and non-holonomic constraint equations. The solution of this problem leads to the direct determination of an initial configuration and velocity that complies with the constraints.

Unfortunately, this requires the implementation of more constraints than required by the regular solution phases, namely the time derivative of the algebraic constraints that depend only on the configuration.

NOTE: Rationale. The rationale of an initial assembly procedure is to drive the problem in a configuration that complies with the kinematic constraints, starting from the configuration that was defined during the input.

Constraint equations can be partitioned in holonomic,

$$\phi(\mathbf{x}, t) = \mathbf{0} \quad (6.6)$$

and non-holonomic, which are usually expressed as linear in the coordinate derivatives,

$$\mathbf{A}(\mathbf{x}, t)\dot{\mathbf{x}} + \mathbf{b}(\mathbf{x}, t) = \mathbf{0} \quad (6.7)$$

It is legitimate that $\phi(\mathbf{x}_0^{(0)}, t_0) \neq \mathbf{0}$, or that $\mathbf{A}(\mathbf{x}_0^{(0)}, t_0)\mathbf{v}_0^{(0)} + \mathbf{b}(\mathbf{x}_0^{(0)}, t_0) \neq \mathbf{0}$, where the subscript 0 indicates the initial condition, and the superscript (0) indicates the values provided at input. In such cases, a procedure is sought that provides \mathbf{x}_0 and \mathbf{v}_0 . Since the problem in many cases is underdetermined (as the number of constraint equations is less than the number of degrees of freedom of the unconstrained problem), a criterion is needed to determine the correction to the input configuration.

A minimum norm correction is obtained by requiring that the solution complies with the constraints and, at the same time, departs as little as possible from the configuration that was provided at input, namely

$$\begin{aligned} & \min_{\mathbf{x}_0, \mathbf{v}_0} \frac{1}{2} \left[(\mathbf{x}_0 - \mathbf{x}_0^{(0)})^T \mathbf{W}_{\mathbf{x}} (\mathbf{x}_0 - \mathbf{x}_0^{(0)}) + (\mathbf{v}_0 - \mathbf{v}_0^{(0)})^T \mathbf{W}_{\mathbf{v}} (\mathbf{v}_0 - \mathbf{v}_0^{(0)}) \right] \\ & \text{subjected to } \begin{aligned} & \dot{\phi}(\mathbf{x}_0, t_0) = \mathbf{0} \\ & \phi = \phi_{/\mathbf{x}}(\mathbf{x}_0, t_0)\mathbf{v}_0 + \phi_{/\mathbf{t}}(\mathbf{x}_0, t_0) = \mathbf{0} \\ & \mathbf{A}(\mathbf{x}_0, t_0)\mathbf{v}_0 + \mathbf{b}(\mathbf{x}_0, t_0) = \mathbf{0} \end{aligned} \end{aligned} \quad (6.8)$$

The problem can be reformulated as

$$\mathbf{W}_{\mathbf{x}} (\mathbf{x}_0 - \mathbf{x}_0^{(0)}) + \phi_{/\mathbf{x}}^T \boldsymbol{\lambda}_{\phi} = \mathbf{0} \quad (6.9a)$$

$$\mathbf{W}_{\mathbf{v}} (\mathbf{v}_0 - \mathbf{v}_0^{(0)}) + \phi_{/\mathbf{x}}^T \boldsymbol{\mu}_{\phi} + \mathbf{A}^T \boldsymbol{\mu}_{\psi} = \mathbf{0} \quad (6.9b)$$

$$\dot{\phi}(\mathbf{x}_0, t_0) = \mathbf{0} \quad (6.9c)$$

$$\phi_{/\mathbf{x}}(\mathbf{x}_0, t_0)\mathbf{v}_0 + \phi_{/\mathbf{t}}(\mathbf{x}_0, t_0) = \mathbf{0} \quad (6.9d)$$

$$\mathbf{A}(\mathbf{x}_0, t_0)\mathbf{v}_0 + \mathbf{b}(\mathbf{x}_0, t_0) = \mathbf{0} \quad (6.9e)$$

After (incomplete) linearization, the problem can be solved as

$$\left[\begin{array}{ccccc} \mathbf{W}_{\mathbf{x}} & \mathbf{0} & \phi_{/\mathbf{x}}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{\mathbf{v}} & \mathbf{0} & \phi_{/\mathbf{x}}^T & \mathbf{A}^T \\ \phi_{/\mathbf{x}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \phi_{/\mathbf{x}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \Delta \mathbf{x}_0 \\ \Delta \mathbf{v}_0 \\ \Delta \boldsymbol{\lambda}_{\phi} \\ \Delta \boldsymbol{\mu}_{\phi} \\ \Delta \boldsymbol{\mu}_{\psi} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{W}_{\mathbf{x}} (\mathbf{x}_0^{(0)} - \mathbf{x}_0) - \phi_{/\mathbf{x}}^T \boldsymbol{\lambda}_{\phi} \\ \mathbf{W}_{\mathbf{v}} (\mathbf{v}_0^{(0)} - \mathbf{v}_0) - \phi_{/\mathbf{x}}^T \boldsymbol{\mu}_{\phi} - \mathbf{A}^T \boldsymbol{\mu}_{\psi} \\ -\dot{\phi} \\ -\dot{\phi} \\ -\mathbf{A}\mathbf{v}_0 - \mathbf{b} \end{array} \right\} \quad (6.10)$$

The problem can be decomposed in two formally independent subproblems,

$$\left[\begin{array}{cc} \mathbf{W}_{\mathbf{x}} & \phi_{/\mathbf{x}}^T \\ \phi_{/\mathbf{x}} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \Delta \mathbf{x}_0 \\ \Delta \boldsymbol{\lambda}_{\phi} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{W}_{\mathbf{x}} (\mathbf{x}_0^{(0)} - \mathbf{x}_0) - \phi_{/\mathbf{x}}^T \boldsymbol{\lambda}_{\phi} \\ -\dot{\phi} \end{array} \right\} \quad (6.11)$$

$$\left[\begin{array}{ccc} \mathbf{W}_{\mathbf{v}} & \phi_{/\mathbf{x}}^T & \mathbf{A}^T \\ \phi_{/\mathbf{x}} & \mathbf{0} & \mathbf{0} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \Delta \mathbf{v}_0 \\ \Delta \boldsymbol{\mu}_{\phi} \\ \Delta \boldsymbol{\mu}_{\psi} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{W}_{\mathbf{v}} (\mathbf{v}_0^{(0)} - \mathbf{v}_0) - \phi_{/\mathbf{x}}^T \boldsymbol{\mu}_{\phi} - \mathbf{A}^T \boldsymbol{\mu}_{\psi} \\ -\dot{\phi} \\ -\mathbf{A}\mathbf{v}_0 - \mathbf{b} \end{array} \right\} \quad (6.12)$$

where the latter depends on the former since ϕ , \mathbf{A} , and \mathbf{b} may depend on \mathbf{x}_0 .

The problems may couple again when, in order to influence the solution, one adds to the right hand side of Eq. (6.9a) some relatively arbitrary contributions that can be interpreted as the residual of an equilibrium,

$$\mathbf{W}_{\mathbf{x}} (\mathbf{x}_0 - \mathbf{x}_0^{(0)}) + \phi_{/\mathbf{x}}^T \boldsymbol{\lambda}_{\phi} = \mathbf{r}(\mathbf{x}_0, \mathbf{v}_0, \mathbf{a}_0, t_0) \quad (6.13)$$

where \mathbf{r} may include position, velocity, and even acceleration-dependent loads. By selectively activating what appears in \mathbf{r} , one would drive the minimization towards a solution that finds the desired trade-off

between staying close to the input configuration and, at the same time, tries to minimize the strain energy, or the gravitational potential energy, or whatever is desired.

For example, one could set $\mathbf{r} = \mathbf{f}(t_0) - \mathbf{K}\mathbf{x}_0$ to account for a dead load $\mathbf{f}(t_0)$, e.g. the weight. By carefully crafting the weight matrix $\mathbf{W}_{\mathbf{x}}$ (e.g. by setting it to zero if the stiffness matrix \mathbf{K} is not singular), one could allow the nodes to depart from their initial configuration and obtain the static equilibrium configuration directly with the initial assembly.

(OUTDATED?) NOTE: this is a work in progress. A new procedure is being considered, which requires only the use of the constraint equation ϕ and its Jacobian matrix $\phi_{/\mathbf{x}}$.

The problem that is solved with the new procedure is:

- solve for configuration and holonomic constraints first

$$\mathbf{K}\mathbf{x} + \phi_{/\mathbf{x}}^T \boldsymbol{\lambda}_\phi = \mathbf{K}\mathbf{x}_0 + \mathbf{f} \quad (6.14)$$

$$\phi(\mathbf{x}, t) = \mathbf{0} \quad (6.15)$$

- then solve for non-holonomic ones,

$$C\mathbf{v} + \mathbf{A}^T \boldsymbol{\lambda}_A = C\mathbf{v}_0 \quad (6.16)$$

$$\mathbf{A}(\mathbf{x}, t)\mathbf{v} + \mathbf{b}(\mathbf{x}, t) = \mathbf{0} \quad (6.17)$$

keeping the configuration \mathbf{x} fixed, so that only the configuration is required to comply with the constraint equations, and the same constraint Jacobian matrix and residual of the regular steps are required;

- finally, after convergence, the following equation is considered:

$$\phi_{/\mathbf{x}}\mathbf{v} + \phi_{/t} = \mathbf{0} \quad (6.18)$$

which uses the Jacobian matrix of the constraint equations; only the derivative of the time-dependent constraints is required.

The number of rows of matrix $\phi_{/\mathbf{x}}$ is equal to the number of degrees of freedom that are constrained, so typically the matrix must be underdetermined. It can be decomposed as

$$\phi_{/\mathbf{x}}^T = \mathbf{Q}\mathbf{R} = [\mathbf{Q}_1 \quad \mathbf{Q}_2] \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \quad (6.19)$$

so

$$\mathbf{Q}_1^T \mathbf{v} + \mathbf{R}_1^{-T} \phi_{/t} = \mathbf{0} \quad (6.20)$$

becomes a compatibility test for the initial velocities. There are two possible choices:

1. the initial configuration assessment fails if the given initial velocities, after the correction occurring during the position and orientation assessment and correction phase, do not pass test (6.20);
2. the initial velocities \mathbf{v} are corrected into \mathbf{v}_c by projecting them in the space that is compatible with the constraints, e.g.:

$$\mathbf{v}_c = \mathbf{v} - \mathbf{Q}_1 (\mathbf{Q}_1^T \mathbf{v} + \mathbf{R}_1^{-T} \phi_{/t}) \quad (6.21)$$

The initial assembly procedure can be repeated until test (6.20) passes.

A consistent implementation of this approach is not available yet; it requires the availability of the time derivatives of the constraint equations, which is an open issue since it is not well understood what is the physical meaning of knowing the time derivatives of a constraint. Implementation issues are still open as well.

6.2 Initial Value Problem

6.2.1 Initial Derivatives

The so-called “derivatives” phase can be thought as computing the initial value of the highest order derivatives at $t = t_0$ before any iteration starts. For simplicity, think of an explicit Ordinary Differential Equation (ODE) problem like

$$\dot{\mathbf{y}} = \hat{\mathbf{f}}(\mathbf{y}, t) \quad (6.22)$$

with the initial value of $t(0) = t_0$ and $\mathbf{y}(0) = \mathbf{y}_0$; then, the computation of $\dot{\mathbf{y}}(0)$ is trivial. Now, the actual problem is Differential Algebraic (DAE) and implicit, i.e. something like

$$\mathbf{f}(\dot{\mathbf{y}}, \mathbf{y}, t) = 0 \quad (6.23)$$

(actually, it's a bit more complicated, since it's index 3), and we still need to compute the derivatives $\dot{\mathbf{y}}(0)$ of the differential variables, and the algebraic variables as well, which in the above representation are hidden in the $\dot{\mathbf{y}}$. During the regular solution phase (i.e. the Newton iteration) we solve a problem of the form

$$\mathbf{f}_{/\dot{\mathbf{y}}} \Delta \dot{\mathbf{y}} + \mathbf{f}_{/\mathbf{y}} \Delta \mathbf{y} = -\mathbf{f} \quad (6.24)$$

and, according to the integration formula that we are using,

$$\Delta \mathbf{y} = \text{dCoef} \cdot \Delta \dot{\mathbf{y}} \quad (6.25)$$

where $\text{dCoef} = b_0 h$ is essentially the time step h times some coefficient b_0 specific for the integration formula; it is 1/2 for Crank-Nicolson, 2/3 for BDF and so on. So the actual iteration is

$$(\mathbf{f}_{/\dot{\mathbf{y}}} + \text{dCoef} \cdot \mathbf{f}_{/\mathbf{y}}) \Delta \dot{\mathbf{y}} = -\mathbf{f}. \quad (6.26)$$

To avoid the need to implement a dedicated routine to compute the initial value of $\dot{\mathbf{y}}$, we iterate over the above reported problem, ideally with a time step of 0, which means that $\mathbf{f}_{/\mathbf{y}}$ is not considered. However, since the problem is differential algebraic, the Jacobian matrix $\mathbf{f}_{/\dot{\mathbf{y}}}$ can be structurally singular, so the time step must be greater than 0, but small enough to let the correction

$$\Delta \mathbf{y} = \text{dCoef} \cdot \Delta \dot{\mathbf{y}} \quad (6.27)$$

be negligible with respect to $\Delta \dot{\mathbf{y}}$.

Assuming the above is true, during this phase the update procedure of the Newton iteration is modified. Instead of computing

$$\dot{\mathbf{y}}^{(i+1)} = \dot{\mathbf{y}}^{(i)} + \Delta \dot{\mathbf{y}} \quad (6.28a)$$

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \text{dCoef} \cdot \Delta \dot{\mathbf{y}}, \quad (6.28b)$$

the “derivatives” update actually consists in

$$\dot{\mathbf{y}}^{(i+1)} = \dot{\mathbf{y}}^{(i)} + \Delta \dot{\mathbf{y}} \quad (6.29a)$$

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)}; \quad (6.29b)$$

namely, the initial value of \mathbf{y} is preserved. This corresponds to performing a Newton iteration with an incomplete Jacobian matrix. For this purpose, nodes and elements that need to take this into account when updating internal states can provide a `DerivativesUpdate` method (declared in `SimulationEntity`).

To conclude, the “derivatives coefficient” defined in the input file is the “`dCoef`” of Eq. (6.26), which can be interpreted as the time step of a “fake” initial step that is used to compute the initial value of the highest order derivatives. It should be as small as possible, but too small makes the problem ill-conditioned (and 0 makes it structurally singular). The default value is usually fine (it rarely needs to be set, unless something really strange is going on).

If the initial derivatives phase requires more than one iteration this may mean that the system is impulsively loaded with inappropriate initial values of states; In this case, the number of iterations can be increased, by setting

```
derivatives coefficient: 1e-9;
derivatives tolerance: 1e-6;
derivatives max iterations: 10;
```

in the `initial value` block. If the solution does not converge, you should enable iterations output, adding

```
output: iterations;
```

in the `initial value` block. If you notice that the error settled to some value, and does not reduce as iterations progress, this means that the incomplete update of Eqs. (6.29) is preventing the problem from converging, because the initial value of \mathbf{y} is compatible with the constraints but does not satisfy equilibrium.

In this case, one can play with the `derivatives coefficient` parameter, to emphasize the “inertia” terms instead of the “elastic” ones, or simply increase the `derivatives tolerance` so that the solution with the current `derivatives coefficient` is considered converged. The latter choice may result in a “rough” behavior of the solution during the initial steps of the simulation.

6.2.2 Dummy Steps

TODO

6.2.3 Regular Steps

TODO

6.3 Inverse Dynamics Problem

In principle, an inverse dynamics problem consists in computing the ‘torques’ (actuator forces) required to comply with equilibrium for a given configuration (position, velocity and acceleration) of the entire system.

In practice, in many cases the motion of the system is not directly known in terms of the coordinates that are used to describe it, but rather in terms of the prescribed motion of the ‘joints’ (the actuators) that apply the unknown torques. As a consequence, an inverse kinematics problem needs to be solved first, to determine the configuration of the system up to the acceleration level as a function of the motion prescribed to the end effector up to the acceleration level. Section 6.3.2 deals with the case of a fully actuated problem, i.e. a problem whose motion is completely prescribed in terms of joint coordinates. Section 6.3.4 deals with the case of an underactuated problem, i.e. a problem whose motion is only partially prescribed in terms of joint coordinates.

In many cases, even the motion of the actuators is not directly known. On the contrary, the motion of another part of the system (called the ‘end effector’) is prescribed, and the actuators’ motion required

to move the end effector as prescribed needs to be computed. As a consequence, the inverse kinematics problem is driven by the motion prescribed to the end effector rather than directly to the joints. This case is specifically dealt with in Section 6.3.5.

6.3.1 Nomenclature

In the following, \mathbf{x} are the n coordinates that describe the motion of the system;

$$\phi(\mathbf{x}) = \mathbf{0} \quad (6.30)$$

are b so-called ‘passive’ constraints, namely those that describe the assembly of the system;

$$\vartheta(\mathbf{x}) = \boldsymbol{\theta} \quad (6.31)$$

are j so-called ‘joint coordinates’, namely those that describe the motion $\boldsymbol{\theta}$ of the joints, with $\phi_{/\mathbf{x}}\vartheta_{/\mathbf{x}}^+ \equiv \mathbf{0}$;

$$\psi(\mathbf{x}) = \boldsymbol{\alpha}(t) \quad (6.32)$$

are c so-called ‘control constraints’, namely equations that prescribe the motion of the end effector, with $\phi_{/\mathbf{x}}\psi_{/\mathbf{x}}^+ \equiv \mathbf{0}$. When the joint motion is directly prescribed, $\boldsymbol{\theta} \equiv \boldsymbol{\alpha}(t)$ and thus $\vartheta(\mathbf{x}) \equiv \psi(\mathbf{x})$. In some cases, a partial overlap may exist between the two sets of equations.

6.3.2 Fully Actuated, Collocated Problem

In this case, $\psi(\mathbf{x}) \equiv \vartheta(\mathbf{x})$ and $c = j = n - b$.

Inverse Kinematics: Position Subproblem

$$\phi(\mathbf{x}) = \mathbf{0} \quad (6.33a)$$

$$\psi(\mathbf{x}) = \boldsymbol{\alpha} \quad (6.33b)$$

using Newton-Raphson:

$$\begin{bmatrix} \phi_{/\mathbf{x}} \\ \psi_{/\mathbf{x}} \end{bmatrix} \Delta \mathbf{x} = \left\{ \begin{array}{l} -\phi(\mathbf{x}) \\ \boldsymbol{\alpha} - \psi(\mathbf{x}) \end{array} \right\}. \quad (6.34)$$

Inverse Kinematics: Velocity Subproblem

$$\phi_{/\mathbf{x}}\dot{\mathbf{x}} = \mathbf{0} \quad (6.35a)$$

$$\psi_{/\mathbf{x}}\dot{\mathbf{x}} = \dot{\boldsymbol{\alpha}} \quad (6.35b)$$

or

$$\begin{bmatrix} \phi_{/\mathbf{x}} \\ \psi_{/\mathbf{x}} \end{bmatrix} \dot{\mathbf{x}} = \left\{ \begin{array}{l} \mathbf{0} \\ \dot{\boldsymbol{\alpha}} \end{array} \right\}. \quad (6.36)$$

Inverse Kinematics: Acceleration Subproblem

$$\phi_{/\mathbf{x}}\ddot{\mathbf{x}} = -(\phi_{/\mathbf{x}}\dot{\mathbf{x}})_{/\mathbf{x}}\dot{\mathbf{x}} \quad (6.37a)$$

$$\psi_{/\mathbf{x}}\ddot{\mathbf{x}} = \ddot{\boldsymbol{\alpha}} - (\psi_{/\mathbf{x}}\dot{\mathbf{x}})_{/\mathbf{x}}\dot{\mathbf{x}} \quad (6.37b)$$

or

$$\begin{bmatrix} \phi_{/\mathbf{x}} \\ \psi_{/\mathbf{x}} \end{bmatrix} \ddot{\mathbf{x}} = \left\{ \begin{array}{l} -(\phi_{/\mathbf{x}}\dot{\mathbf{x}})_{/\mathbf{x}}\dot{\mathbf{x}} \\ \ddot{\boldsymbol{\alpha}} - (\psi_{/\mathbf{x}}\dot{\mathbf{x}})_{/\mathbf{x}}\dot{\mathbf{x}} \end{array} \right\}. \quad (6.38)$$

Inverse Dynamics Subproblem

$$M\ddot{x} + \phi_{/x}^T \lambda + \psi_{/x}^T c = f \quad (6.39)$$

or

$$\begin{bmatrix} \phi_{/x} \\ \psi_{/x} \end{bmatrix}^T \begin{Bmatrix} \lambda \\ c \end{Bmatrix} = f - M\ddot{x}. \quad (6.40)$$

The matrix of the Newton-Raphson problem for the position is identical to the matrices of the linear problems for the velocity and the acceleration, while the inverse dynamics problem uses its transpose, so the same factorization can be easily reused.

6.3.3 Fully Actuated, Non-Collocated Problem

In this case, $\psi(x) \neq \vartheta(x)$, but still $c = j = n - b$.

Inverse Kinematics: Position, Velocity and Acceleration Subproblems

The position, velocity and acceleration subproblems that define the inverse kinematics problem are identical to those of the collocated case of Section 6.3.2.

Inverse Dynamics Subproblem

$$M\ddot{x} + \phi_{/x}^T \lambda + \vartheta_{/x}^T c = f \quad (6.41)$$

or

$$\begin{bmatrix} \phi_{/x} \\ \vartheta_{/x} \end{bmatrix}^T \begin{Bmatrix} \lambda \\ c \end{Bmatrix} = f - M\ddot{x}. \quad (6.42)$$

The inverse dynamics problem uses a different matrix from the transpose of that that was used for the inverse kinematics subproblems.

6.3.4 Underdetermined, Underactuated but Collocated Problem

In this case, again $\psi(x) \equiv \vartheta(x)$ but $c = j < n - b$.

Inverse Kinematics: Position Subproblem

Constraint equations:

$$\phi(x) = 0 \quad (6.43a)$$

$$\psi(x) = \alpha(t) \quad (6.43b)$$

with $\phi_{/x}$ and $\psi_{/x}$ rectangular, full row rank. The problem is underdetermined; as a consequence, some criteria are needed to find an optimal solution.

The problem can be augmented by

$$K(x - x_0) + \phi_{/x}^T \lambda + \psi_{/x}^T \mu = 0 \quad (6.44)$$

This nonlinear problem is solved for \boldsymbol{x} , $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ to convergence (\boldsymbol{K} , $\phi_{/\boldsymbol{x}}$ and $\psi_{/\boldsymbol{x}}$ may further depend on \boldsymbol{x}). It corresponds to a least-squares solution for $\boldsymbol{x} - \boldsymbol{x}_0$, where the quadratic form

$$J_{\boldsymbol{x}} = \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}_0)^T \boldsymbol{K} (\boldsymbol{x} - \boldsymbol{x}_0) \quad (6.45)$$

is minimized subjected to $\boldsymbol{\phi} = \mathbf{0}$ and $\boldsymbol{\psi} = \boldsymbol{\alpha}$, and weighted by matrix \boldsymbol{K} ; \boldsymbol{x}_0 is a reference solution. Matrix \boldsymbol{K} can further depend on \boldsymbol{x} .

The reference solution \boldsymbol{x}_0 can be used to further control the quality of the solution. For example, it may represent a prescribed tentative, although possibly incompatible, trajectory. Another option consists in augmenting $J_{\boldsymbol{x}}$ with another quadratic form

$$J_{\boldsymbol{x}} = \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}_0)^T \boldsymbol{K} (\boldsymbol{x} - \boldsymbol{x}_0) + w_{\boldsymbol{x}} \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}_{\text{prev}})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{x}_{\text{prev}}) \quad (6.46)$$

where $\boldsymbol{x}_{\text{prev}}$ is the value of \boldsymbol{x} at the previous time step. This modified quadratic form weights the rate of change of \boldsymbol{x} within two consecutive steps. As a consequence, minimal position changes (i.e. velocities), weighted by the mass of the system, are sought. The corresponding problem is

$$\boldsymbol{K} (\boldsymbol{x} - \boldsymbol{x}_0) + w_{\boldsymbol{x}} \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{x}_{\text{prev}}) + \phi_{/\boldsymbol{x}}^T \boldsymbol{\lambda} + \psi_{/\boldsymbol{x}}^T \boldsymbol{\mu} = \mathbf{0} \quad (6.47a)$$

$$\boldsymbol{\phi} (\boldsymbol{x}) = \mathbf{0} \quad (6.47b)$$

$$\boldsymbol{\psi} (\boldsymbol{x}) = \boldsymbol{\alpha}, \quad (6.47c)$$

or

$$(\boldsymbol{K} + w_{\boldsymbol{x}} \boldsymbol{M}) \boldsymbol{x} + \phi_{/\boldsymbol{x}}^T \boldsymbol{\lambda} + \psi_{/\boldsymbol{x}}^T \boldsymbol{\mu} = \boldsymbol{K} \boldsymbol{x}_0 + w_{\boldsymbol{x}} \boldsymbol{M} \boldsymbol{x}_{\text{prev}} \quad (6.48a)$$

$$\boldsymbol{\phi} (\boldsymbol{x}) = \mathbf{0} \quad (6.48b)$$

$$\boldsymbol{\psi} (\boldsymbol{x}) = \boldsymbol{\alpha}. \quad (6.48c)$$

Using Newton-Raphson:

$$\left[\begin{array}{ccc} \boldsymbol{K} + w_{\boldsymbol{x}} \boldsymbol{M} & \phi_{/\boldsymbol{x}}^T & \psi_{/\boldsymbol{x}}^T \\ \phi_{/\boldsymbol{x}} & \mathbf{0} & \mathbf{0} \\ \psi_{/\boldsymbol{x}} & \mathbf{0} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \Delta \boldsymbol{x} \\ \Delta \boldsymbol{\lambda} \\ \Delta \boldsymbol{\mu} \end{array} \right\} = \left\{ \begin{array}{c} \boldsymbol{K} (\boldsymbol{x}_0 - \boldsymbol{x}) + w_{\boldsymbol{x}} \boldsymbol{M} (\boldsymbol{x}_{\text{prev}} - \boldsymbol{x}) - \phi_{/\boldsymbol{x}}^T \boldsymbol{\lambda} - \psi_{/\boldsymbol{x}}^T \boldsymbol{\mu} \\ -\boldsymbol{\phi} (\boldsymbol{x}) \\ \boldsymbol{\alpha}(t) - \boldsymbol{\psi} (\boldsymbol{x}) \end{array} \right\} \quad (6.49)$$

Inverse Kinematics: Velocity Subproblem

Constraint first derivative:

$$\boldsymbol{R} (\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_0) + \phi_{/\boldsymbol{x}}^T \boldsymbol{\lambda} + \psi_{/\boldsymbol{x}}^T \boldsymbol{\mu} = \mathbf{0} \quad (6.50a)$$

$$\phi_{/\boldsymbol{x}} \dot{\boldsymbol{x}} = \mathbf{0} \quad (6.50b)$$

$$\psi_{/\boldsymbol{x}} \dot{\boldsymbol{x}} = \dot{\boldsymbol{\alpha}}. \quad (6.50c)$$

In analogy with the position constraint case, it corresponds to minimizing the quadratic form

$$J_{\dot{\boldsymbol{x}}} = \frac{1}{2} (\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_0)^T \boldsymbol{R} (\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_0), \quad (6.51)$$

which can be augmented as well, resulting in

$$J_{\dot{\boldsymbol{x}}} = \frac{1}{2} (\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_0)^T \boldsymbol{R} (\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_0) + w_{\dot{\boldsymbol{x}}} \frac{1}{2} (\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_{\text{prev}})^T \boldsymbol{M} (\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_{\text{prev}}), \quad (6.52)$$

to minimize the velocity increment between two consecutive time steps. The corresponding problem is

$$(\mathbf{R} + w_{\dot{\mathbf{x}}} \mathbf{M}) \dot{\mathbf{x}} + \boldsymbol{\phi}_{/\mathbf{x}}^T \boldsymbol{\lambda} + \boldsymbol{\psi}_{/\mathbf{x}}^T \boldsymbol{\mu} = \mathbf{R} \dot{\mathbf{x}}_0 + w_{\dot{\mathbf{x}}} \mathbf{M} \dot{\mathbf{x}}_{\text{prev}} \quad (6.53a)$$

$$\boldsymbol{\phi}_{/\mathbf{x}} \dot{\mathbf{x}} = \mathbf{0} \quad (6.53b)$$

$$\boldsymbol{\psi}_{/\mathbf{x}} \dot{\mathbf{x}} = \dot{\boldsymbol{\alpha}}, \quad (6.53c)$$

or

$$\begin{bmatrix} \mathbf{R} + w_{\dot{\mathbf{x}}} \mathbf{M} & \boldsymbol{\phi}_{/\mathbf{x}}^T & \boldsymbol{\psi}_{/\mathbf{x}}^T \\ \boldsymbol{\phi}_{/\mathbf{x}} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\psi}_{/\mathbf{x}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{x}} \\ \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{Bmatrix} = \begin{Bmatrix} \mathbf{R} \dot{\mathbf{x}}_0 + w_{\dot{\mathbf{x}}} \mathbf{M} \dot{\mathbf{x}}_{\text{prev}} \\ \mathbf{0} \\ \dot{\boldsymbol{\alpha}} \end{Bmatrix}. \quad (6.54)$$

The constraint derivative problem is linear in $\dot{\mathbf{x}}$; the same symbols $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are used for the multipliers since they are ineffective (their value is never used). The problems can be solved sequentially. Only in case $\mathbf{K} \equiv \mathbf{R}$ and $w_{\mathbf{x}} \equiv w_{\dot{\mathbf{x}}}$ both problems use the same matrix and thus the same factorization can be reused.

Inverse Dynamics Subproblem

The second derivative of the constraint cannot be resolved as in the fully actuated case, otherwise it could yield accelerations that cannot be imposed by the constraints. On the contrary, the inverse dynamics problem of Eq. (6.39) is directly solved, yielding both the accelerations and the torques,

$$\mathbf{M} \ddot{\mathbf{x}} + \boldsymbol{\phi}_{/\mathbf{x}} \boldsymbol{\lambda} + \boldsymbol{\psi}_{/\mathbf{x}} \mathbf{c} = \mathbf{f} \quad (6.55a)$$

$$\boldsymbol{\phi}_{/\mathbf{x}} \ddot{\mathbf{x}} = -(\boldsymbol{\phi}_{/\mathbf{x}} \dot{\mathbf{x}})_{/\mathbf{x}} \dot{\mathbf{x}} \quad (6.55b)$$

$$\boldsymbol{\psi}_{/\mathbf{x}} \ddot{\mathbf{x}} = \dot{\boldsymbol{\alpha}} - (\boldsymbol{\psi}_{/\mathbf{x}} \dot{\mathbf{x}})_{/\mathbf{x}} \dot{\mathbf{x}} \quad (6.55c)$$

or

$$\begin{bmatrix} \mathbf{M} & \boldsymbol{\phi}_{/\mathbf{x}}^T & \boldsymbol{\psi}_{/\mathbf{x}}^T \\ \boldsymbol{\phi}_{/\mathbf{x}} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\psi}_{/\mathbf{x}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{x}} \\ \boldsymbol{\lambda} \\ \mathbf{c} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ -(\boldsymbol{\phi}_{/\mathbf{x}} \dot{\mathbf{x}})_{/\mathbf{x}} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\alpha}} - (\boldsymbol{\psi}_{/\mathbf{x}} \dot{\mathbf{x}})_{/\mathbf{x}} \dot{\mathbf{x}} \end{Bmatrix} \quad (6.56)$$

As a consequence, the same structure of the inverse kinematic problems is achieved, and the last problem of Eq. (6.56) directly yields both accelerations and multipliers.

6.3.5 Underdetermined, Overcontrolled Problem

In this case, $\boldsymbol{\psi}(\mathbf{x}) \neq \boldsymbol{\vartheta}(\mathbf{x})$, $c < n - b$, $j = n - b$, which implies $c < j$. This is the case, for example, of an inverse biomechanics problem, where each degree of freedom is a joint commanded by a set of muscles. The inverse kinematics problem is solved by prescribing the motion of some part (e.g. a hand or a foot) in order to determine an ‘optimal’ (e.g. in terms of maximal ergonomics) motion. Then, an inverse dynamics problem is computed by freeing the part whose motion was initially prescribed, and by computing the torque required by all joints.

Inverse Kinematics: Position and Velocity Subproblems

The constraint and its first derivative are dealt with as in Section 6.3.4.

Inverse Kinematics: Acceleration Subproblem

Constraint second derivative:

$$\mathbf{M}(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_0) + \boldsymbol{\phi}_{/\mathbf{x}}^T \boldsymbol{\lambda} + \boldsymbol{\psi}_{/\mathbf{x}}^T \boldsymbol{\mu} = \mathbf{0} \quad (6.57a)$$

$$\boldsymbol{\phi}_{/\mathbf{x}} \ddot{\mathbf{x}} = -(\boldsymbol{\phi}_{/\mathbf{x}} \dot{\mathbf{x}})_{/\mathbf{x}} \dot{\mathbf{x}} \quad (6.57b)$$

$$\boldsymbol{\psi}_{/\mathbf{x}} \ddot{\mathbf{x}} = \ddot{\boldsymbol{\alpha}} - (\boldsymbol{\psi}_{/\mathbf{x}} \dot{\mathbf{x}})_{/\mathbf{x}} \dot{\mathbf{x}}. \quad (6.57c)$$

In analogy with the position constraint case, it corresponds to minimizing the quadratic form

$$J_{\ddot{\mathbf{x}}} = \frac{1}{2} (\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_0)^T \mathbf{M} (\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_0), \quad (6.58)$$

which can be augmented as well, resulting in

$$J_{\ddot{\mathbf{x}}} = \frac{1}{2} (\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_0)^T \mathbf{M} (\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_0) + w_{\ddot{\mathbf{x}}} \frac{1}{2} (\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_{\text{prev}})^T \mathbf{M} (\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_{\text{prev}}), \quad (6.59)$$

to minimize the acceleration increment between two consecutive time steps. The corresponding problem is

$$(1 + w_{\ddot{\mathbf{x}}}) \mathbf{M} \ddot{\mathbf{x}} + \boldsymbol{\phi}_{/\mathbf{x}}^T \boldsymbol{\lambda} + \boldsymbol{\psi}_{/\mathbf{x}}^T \boldsymbol{\mu} = \mathbf{M} \ddot{\mathbf{x}}_0 + w_{\ddot{\mathbf{x}}} \mathbf{M} \ddot{\mathbf{x}}_{\text{prev}} \quad (6.60a)$$

$$\boldsymbol{\phi}_{/\mathbf{x}} \ddot{\mathbf{x}} = -(\boldsymbol{\phi}_{/\mathbf{x}} \dot{\mathbf{x}})_{/\mathbf{x}} \dot{\mathbf{x}} \quad (6.60b)$$

$$\boldsymbol{\psi}_{/\mathbf{x}} \ddot{\mathbf{x}} = \ddot{\boldsymbol{\alpha}} - (\boldsymbol{\psi}_{/\mathbf{x}} \dot{\mathbf{x}})_{/\mathbf{x}} \dot{\mathbf{x}}, \quad (6.60c)$$

or

$$\begin{bmatrix} (1 + w_{\ddot{\mathbf{x}}}) \mathbf{M} & \boldsymbol{\phi}_{/\mathbf{x}}^T & \boldsymbol{\psi}_{/\mathbf{x}}^T \\ \boldsymbol{\phi}_{/\mathbf{x}} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\psi}_{/\mathbf{x}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{x}} \\ \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{Bmatrix} = \begin{Bmatrix} \mathbf{M} \ddot{\mathbf{x}}_0 + w_{\ddot{\mathbf{x}}} \mathbf{M} \ddot{\mathbf{x}}_{\text{prev}} \\ -(\boldsymbol{\phi}_{/\mathbf{x}} \dot{\mathbf{x}})_{/\mathbf{x}} \dot{\mathbf{x}} \\ \ddot{\boldsymbol{\alpha}} - (\boldsymbol{\psi}_{/\mathbf{x}} \dot{\mathbf{x}})_{/\mathbf{x}} \dot{\mathbf{x}} \end{Bmatrix}. \quad (6.61)$$

Inverse Dynamics Subproblem

The inverse dynamics subproblem is identical to that of the fully determined, non-collocated case, Eq. (6.41).

Implementation

- during ‘position’ inverse kinematics substep:
 - assemble $\boldsymbol{\phi}_{/\mathbf{x}}$ and $\boldsymbol{\phi}_{/\mathbf{x}}^T$ for passive constraints
 - assemble $\boldsymbol{\psi}_{/\mathbf{x}}$ and $\boldsymbol{\psi}_{/\mathbf{x}}^T$ for prescribed motion
 - assemble equations related to $\boldsymbol{\vartheta}_{/\mathbf{x}}$ as $\mathbf{c} = \mathbf{0}$ to neutralize them
 - assemble dummy springs
 - optionally assemble mass matrix contribution weighted by $w_{/\mathbf{x}}$
- during ‘velocity’ inverse kinematics substep:
 - assemble $\boldsymbol{\phi}_{/\mathbf{x}}$ and $\boldsymbol{\phi}_{/\mathbf{x}}^T$ for passive constraints
 - assemble $\boldsymbol{\psi}_{/\mathbf{x}}$ and $\boldsymbol{\psi}_{/\mathbf{x}}^T$ for prescribed motion

- assemble equations related to $\vartheta_{/\alpha}$ as $c = \mathbf{0}$ to neutralize them
 - assemble dummy dampers
 - optionally assemble mass matrix contribution weighted by $w_{/\dot{\alpha}}$
- during ‘acceleration’ inverse kinematics substep:
 - assemble $\phi_{/\alpha}$ and $\phi_{/\alpha}^T$ for passive constraints
 - assemble $\psi_{/\alpha}$ and $\psi_{/\alpha}^T$ for prescribed motion
 - assemble equations related to $\vartheta_{/\alpha}$ as $c = \mathbf{0}$ to neutralize them
 - assemble mass matrix contribution, optionally weighted by $(1 + w_{/\ddot{\alpha}})$
- during inverse dynamics substep:
 - assemble $\phi_{/\alpha}$ and $\phi_{/\alpha}^T$ for passive constraints
 - assemble $\vartheta_{/\alpha}$ and $\vartheta_{/\alpha}^T$ for torques
 - assemble equations related to $\psi_{/\alpha}$ as $\mu = \mathbf{0}$ to neutralize them
 - assemble other elements (external forces, springs, etc.)

Chapter 7

Data Structure

(This chapter is a mess.)

7.1 Constitutive Laws

Code that uses constitutive laws requires

$$\mathbf{f} = \mathbf{f}(\boldsymbol{\epsilon}, \dot{\boldsymbol{\epsilon}}, t, \dots) \quad (7.1)$$

where \mathbf{f} can be 1, 3, and 6 dimensional. Similarly, $\boldsymbol{\epsilon}$ and $\dot{\boldsymbol{\epsilon}}$ respectively are 1, 3, and 6 dimensional (scalar, `Vec3`, and `Vec6`).

The `ConstitutiveLaw` provides

$$\text{GetF}() = \mathbf{f}(\boldsymbol{\epsilon}, \dot{\boldsymbol{\epsilon}}, t, \dots) \quad (7.2a)$$

$$\text{GetFDE}() = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\epsilon}} \quad (7.2b)$$

$$\text{GetFDEPrime}() = \frac{\partial \mathbf{f}}{\partial \dot{\boldsymbol{\epsilon}}} \quad (7.2c)$$

The values returned by these methods are updated by a call to `Update()`.

7.2 ExpandableRowVector

`ExpandableRowVector` is a class that supports the computation of nested Jacobian matrices.

Consider a scalar y that depends on a set of variables \mathbf{k} , namely $y = y(\mathbf{k})$. The generic variable k_i , in turn, may be directly a subset of the variables of the problem, \mathbf{x} , with $k_i = x_j$, or depend on them either directly, namely $k_i = k_i(\mathbf{x})$, or indirectly, namely $k_i = k_i(\mathbf{k}'(\mathbf{x}))$ and so on, with as many levels of recursion as needed, namely

$$\frac{\partial y}{\partial x_j} = \sum_i \frac{\partial y}{\partial k_i} \left(\sum_{i'} \frac{\partial k_i}{\partial k'_{i'}} \left(\cdots \sum_j \frac{\partial k^{(n)}_{i^{(n)}}}{\partial x_j} \right) \right). \quad (7.3)$$

The computation of the contribution of y to the Jacobian matrix of the problem requires the capability to assemble the partial derivatives of y with respect to the variables \mathbf{x} in the appropriate locations.

The `ExpandableRowVector` allows to associate each partial derivative of y with respect to the variables \mathbf{k} it depends on, namely $y_{/k_i}$, to either

- a) its index, if k_i is directly a variable of the problem, $k_i \rightarrow x_j$, or
- b) to another `ExpandableRowVector` that contains the partial derivatives of the corresponding k_i with respect to the variables it depends on.

Since the assembly of the contributions to the Jacobian matrix of the problem is usually done using submatrices, within an element the association between a coefficient and a problem variable is not done based on the absolute numbering of the variable, but rather on a local numbering within the submatrix. The submatrix in turn will take care of mapping local indexing to global indexing when it is assembled to the global Jacobian matrix of the problem.

When the variable k_i directly corresponds to the problem variable x_j , the `ExpandableRowVector` provides a method to associate one of its elements to the problem variable indexing within the submatrix. So, if $k_i \rightarrow x_j$, and the global index j corresponds to index `ip` in the submatrix,

```
v.Set(y_{k_i}, i, ip)
```

simultaneously sets the value y_{k_i} and the variable subindex `ip` of the `i`-th element of vector `v`.

Alternatively, the method

```
v.SetIdx(i, ip)
```

associates the variable subindex `ip` to the `i`-th element of the vector.

When k_i is a function of a set of variables, and its partial derivatives with respect to those variables are stored in another `ExpandableRowVector` `w`, the method

```
v.Link(i, &w)
```

allows to associate vector `w` to the `i`-th element of vector `v`.

As soon as `v` is assembled, it can be contributed to a specific equation (row) of a Jacobian matrix.

For example, calling the method

```
void
Add(FullSubMatrixHandler& WM,
    const integer eq,
    const doublereal c = 1.) const
```

of vector `v` corresponds to

```
for (integer j = 1; j <= n; j++) {
    WM(eq, idx(j)) += c*v(j);
}
```

where `idx` is the vector that contains the submatrix indexes of the elements of `v`, as set by the `SetIdx()` method.

If another `ExpandableRowVector` is linked to the `i`-th element of vector `v`, the related operation on the submatrix is propagated recursively as

```
for (integer j = 1; j <= n; j++) {
    if ( /* linked to another ExpandableRowVector */ ) {
        // propagate operation
        xm(j)->Add(WM, eq, c*v(j));
    } else {
        WM(eq, idx(j)) += c*v(j);
    }
}
```

where `xm` is the container of the pointers to the linked `ExpandableRowVector`.

Blocks of equations can be added simultaneously using the method

```
void
Add(FullSubMatrixHandler& WM,
    const std::vector<integer>& eq,
    const std::vector<doublereal>& cc,
    const doublereal c = 1.) const
```

The same operation illustrated above is performed on each equation whose subindex is stored in the array `eq`, weighed by the coefficients stored in the array `cc`.

A similar method allows to build a subvector. Its use is currently undocumented.

Example.

```
ExpandableRowVector v(4);
ExpandableRowVector w(2);
FullSubMatrixHandler m(6, 9);

w.Set(10., 1, 8);
w.Set(20., 2, 9);

v.Set(1., 1, 4);
v.Set(2., 2, 5);
v.Set(3., 3, 6);
v.Set(10., 4);
v.Link(4, &w);

v.Add(m, 5, 1.);

//      col.  1   2   3   4   5   6   7   8   9
// m(5, :) = { 0., 0., 0., 1., 2., 3., 0., 100., 200. }

std::vector<integer> eq(3);
std::vector<doublereal> cc(3);
eq[0] = 1; cc[0] = 1.;
eq[1] = 2; cc[1] = 2.;
eq[2] = 3; cc[2] = 3.;

v.Add(m, eq, cc, 1.);

//      col.  1   2   3   4   5   6   7   8   9
// m(1, :) = { 0., 0., 0., 1., 2., 3., 0., 100., 200. }
// m(2, :) = { 0., 0., 0., 2., 4., 6., 0., 200., 400. }
// m(3, :) = { 0., 0., 0., 3., 6., 9., 0., 300., 600. }
```

Chapter 8

Nodes

8.1 Structural Nodes

Structural nodes provide the kinematics unknowns and the corresponding equilibrium equations.

8.1.1 Dynamic Structural Nodes

The dynamic structural node also provides momentum and momenta moment unknowns, and the equations that represent their definition in terms of the inertia properties and the node's kinematics:

$$\begin{aligned}\boldsymbol{\beta} &= m\dot{\boldsymbol{x}}_{CM} \\ &= m(\dot{\boldsymbol{x}} + \boldsymbol{\omega} \times \mathbf{b}) \\ &= m\dot{\boldsymbol{x}} + \boldsymbol{\omega} \times \mathbf{s}\end{aligned}\tag{8.1a}$$

$$\begin{aligned}\boldsymbol{\gamma} &= \boldsymbol{\gamma}_{CM} + \mathbf{b} \times \boldsymbol{\beta} \\ &= \mathbf{J}_{CM}\boldsymbol{\omega} + \mathbf{b} \times \boldsymbol{\beta} \\ &= \mathbf{s} \times \dot{\boldsymbol{x}} + \mathbf{J}\boldsymbol{\omega},\end{aligned}\tag{8.1b}$$

where

$$\mathbf{b} = \frac{\mathbf{s}}{m}\tag{8.2}$$

is the distance between the node and the CM of the body, which is constant in the reference frame of the node for a rigid body, and $\mathbf{J} = \mathbf{J}_{CM} + m\mathbf{b} \times \mathbf{b} \times ^T$.

The contribution to the equations of motion (Newton-Euler) is

$$\mathbf{f}_{in} = \dot{\boldsymbol{\beta}}\tag{8.3a}$$

$$\begin{aligned}\mathbf{m}_{in} &= \dot{\boldsymbol{\gamma}}_{CM} + \mathbf{b} \times \mathbf{f}_{in} \\ &= \dot{\boldsymbol{\gamma}} - (\boldsymbol{\omega} \times \mathbf{b}) \times \boldsymbol{\beta} \\ &= \dot{\boldsymbol{\gamma}} - m(\boldsymbol{\omega} \times \mathbf{b}) \times (\dot{\boldsymbol{x}} + \boldsymbol{\omega} \times \mathbf{b}) \\ &= \dot{\boldsymbol{\gamma}} + m\dot{\boldsymbol{x}} \times (\boldsymbol{\omega} \times \mathbf{b}) \\ &= \dot{\boldsymbol{\gamma}} + m\dot{\boldsymbol{x}} \times (\dot{\boldsymbol{x}} + \boldsymbol{\omega} \times \mathbf{b}) \\ &= \dot{\boldsymbol{\gamma}} + \dot{\boldsymbol{x}} \times \boldsymbol{\beta}\end{aligned}\tag{8.3b}$$

The perturbation of the momentum and momenta moment definitions yields

$$\delta\beta = m\delta\dot{x} - \mathbf{s} \times \delta\omega - \omega \times \mathbf{s} \times \theta_\delta \quad (8.4a)$$

$$\delta\gamma = \mathbf{s} \times \delta\dot{x} + \mathbf{J}\delta\omega + (\dot{x} \times \mathbf{s} \times - (\mathbf{J}\omega) \times + \mathbf{J}\omega \times) \theta_\delta \quad (8.4b)$$

The perturbation of the inertia force and moment yields

$$\delta f_{in} = \delta\dot{\beta} \quad (8.5a)$$

$$\delta m_{in} = \delta\dot{\gamma} - \beta \times \delta\dot{x} + \dot{x} \times \delta\beta \quad (8.5b)$$

Accelerations

Since momentum and momenta moments are used as the unknowns that take care of inertia, linear and angular accelerations are not directly available from the simulation. They are computed, on demand, as postprocessing from the derivatives of the definitions of the momentum and the momenta moment.

The differentiation of Equations (8.1) yields

$$\dot{\beta} = m\ddot{x} + \dot{\omega} \times \mathbf{s} + \omega \times \omega \times \mathbf{s} \quad (8.6a)$$

$$\dot{\gamma} = (\omega \times \mathbf{s}) \times \dot{x} + \mathbf{s} \times \ddot{x} + \omega \times \mathbf{J}\omega + \mathbf{J}\dot{\omega}. \quad (8.6b)$$

As a consequence, the linear and angular accelerations are computed as

$$\dot{\omega} = \mathbf{J}_{CM}^{-1} (\dot{\gamma} - \mathbf{b} \times \dot{\beta} + \dot{x} \times \beta - \omega \times \mathbf{J}_{CM}\omega) \quad (8.7)$$

$$\begin{aligned} \ddot{x} &= \frac{1}{m} (\dot{\beta} - \dot{\omega} \times \mathbf{s} - \omega \times \omega \times \mathbf{s}) \\ &= \left(\frac{1}{m} \mathbf{I} + \mathbf{b} \times^T \mathbf{J}_{CM}^{-1} \mathbf{b} \times \right) \dot{\beta} - \mathbf{b} \times^T \mathbf{J}_{CM}^{-1} \dot{\gamma} - \mathbf{b} \times^T \mathbf{J}_{CM}^{-1} (\dot{x} \times \beta - \omega \times \mathbf{J}_{CM}\omega), \end{aligned} \quad (8.8)$$

namely

$$\begin{Bmatrix} \ddot{x} \\ \dot{\omega} \end{Bmatrix} = \mathbf{M}^{-1} \begin{Bmatrix} \dot{\beta} \\ \dot{\gamma} \end{Bmatrix} + \begin{bmatrix} -\mathbf{b} \times^T \\ \mathbf{I} \end{bmatrix} \mathbf{J}_{CM}^{-1} (\dot{x} \times \beta - \omega \times \mathbf{J}_{CM}\omega) \quad (8.9)$$

Note on the inverse of the mass matrix

The mass matrix is

$$\mathbf{M} = \begin{bmatrix} m\mathbf{I} & \mathbf{s} \times^T \\ \mathbf{s} \times & \mathbf{J} \end{bmatrix} \quad (8.10)$$

Its inverse is

$$\begin{aligned} \mathbf{M}^{-1} &= \begin{bmatrix} (m\mathbf{I} - \mathbf{s} \times^T \mathbf{J}^{-1} \mathbf{s} \times)^{-1} & -\frac{1}{m} \mathbf{s} \times^T \left(\mathbf{J} - \frac{1}{m} \mathbf{s} \times \mathbf{s} \times^T \right)^{-1} \\ -\frac{1}{m} \left(\mathbf{J} - \frac{1}{m} \mathbf{s} \times \mathbf{s} \times^T \right)^{-1} \mathbf{s} \times & \left(\mathbf{J} - \frac{1}{m} \mathbf{s} \times \mathbf{s} \times^T \right)^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{m} \mathbf{I} + \mathbf{b} \times^T \mathbf{J}_{CM}^{-1} \mathbf{b} \times & -\mathbf{b} \times^T \mathbf{J}_{CM}^{-1} \\ -\mathbf{J}_{CM}^{-1} \mathbf{b} \times & \mathbf{J}_{CM}^{-1} \end{bmatrix}. \end{aligned} \quad (8.11)$$

In fact, according to the matrix inversion lemma,

$$(m\mathbf{I} - \mathbf{s} \times {}^T \mathbf{J}^{-1} \mathbf{s} \times)^{-1} = \frac{1}{m} \mathbf{I} + \mathbf{b} \times {}^T \mathbf{J}_{CM}^{-1} \mathbf{b} \times \quad (8.12)$$

(NOTE: simplifications when using updated-updated formulas).

Perturbation of momentum and momenta moment derivatives

According to Eqs. (8.4), the perturbation of the linear and angular velocity is

$$\begin{Bmatrix} \delta \dot{\mathbf{x}} \\ \delta \omega \end{Bmatrix} = \mathbf{M}^{-1} \left(\begin{Bmatrix} \delta \beta \\ \delta \gamma \end{Bmatrix} - \begin{bmatrix} -\omega \times \mathbf{s} \times \\ \dot{\mathbf{x}} \times \mathbf{s} \times - (\mathbf{J} \omega) \times + \mathbf{J} \omega \times \end{bmatrix} \theta_\delta \right) \quad (8.13a)$$

The perturbation of the momentum and momenta moment derivatives is

$$\delta \dot{\beta} = m \delta \ddot{\mathbf{x}} - \mathbf{s} \times \delta \dot{\omega} - (\omega \times \mathbf{s}) \times \delta \omega - \omega \times \mathbf{s} \times \delta \omega - \dot{\omega} \times \mathbf{s} \times \theta_\delta - \omega \times \omega \times \mathbf{s} \times \theta_\delta \quad (8.14a)$$

$$\begin{aligned} \delta \dot{\gamma} &= \mathbf{s} \times \delta \ddot{\mathbf{x}} + \mathbf{J} \delta \dot{\omega} + (\omega \times \mathbf{s}) \times \delta \dot{\mathbf{x}} + \dot{\mathbf{x}} \times \mathbf{s} \times \delta \omega + \omega \times \mathbf{J} \delta \omega - (\mathbf{J} \omega) \times \delta \omega \\ &\quad + \dot{\mathbf{x}} \times \omega \times \mathbf{s} \times \theta_\delta + \ddot{\mathbf{x}} \times \mathbf{s} \times \theta_\delta - \omega \times (\mathbf{J} \omega) \times \theta_\delta + \omega \times \mathbf{J} \omega \times \theta_\delta - (\mathbf{J} \dot{\omega}) \times \theta_\delta + \mathbf{J} \dot{\omega} \times \theta_\delta. \end{aligned} \quad (8.14b)$$

As a consequence,

$$\begin{aligned} \begin{Bmatrix} \delta \ddot{\mathbf{x}} \\ \delta \dot{\omega} \end{Bmatrix} &= \mathbf{M}^{-1} \left(\begin{Bmatrix} \dot{\beta} \\ \dot{\gamma} \end{Bmatrix} - \begin{bmatrix} \mathbf{0} \\ (\omega \times \mathbf{s}) \times \end{bmatrix} \delta \dot{\mathbf{x}} - \begin{bmatrix} -(\omega \times \mathbf{s}) \times - \omega \times \mathbf{s} \times \\ \dot{\mathbf{x}} \times \mathbf{s} \times + \omega \times \mathbf{J} - (\mathbf{J} \omega) \times \end{bmatrix} \delta \omega \right. \\ &\quad \left. - \begin{bmatrix} -\dot{\omega} \times \mathbf{s} \times - \omega \times \omega \times \mathbf{s} \times \\ \dot{\mathbf{x}} \times \omega \times \mathbf{s} \times + \ddot{\mathbf{x}} \times \mathbf{s} \times - \omega \times (\mathbf{J} \omega) \times + \omega \times \mathbf{J} \omega \times - (\mathbf{J} \dot{\omega}) \times + \mathbf{J} \dot{\omega} \times \end{bmatrix} \theta_\delta \right) \end{aligned} \quad (8.15)$$

(NOTE: simplifications when using updated-updated formulas).

Variable Mass Momentum:

$$\beta = m \dot{\mathbf{x}}_{CM} \quad (8.16)$$

with

$$\dot{\mathbf{x}}_{CM} = \dot{\mathbf{x}} + \omega \times \mathbf{b} + \mathbf{R} \dot{\tilde{\mathbf{b}}}. \quad (8.17)$$

$$\dot{\beta} = m \ddot{\mathbf{x}}_{CM} + \dot{m} \dot{\mathbf{x}}_{CM} \quad (8.18)$$

Eq. (8.3a) becomes

$$\mathbf{f}_{in} = \dot{\beta} - \dot{m} \dot{\mathbf{x}}_{CM}. \quad (8.19)$$

Only the portion of $\dot{\beta}$ related to the change in mass is subtracted from the inertia forces; the portion related to the change in mass distribution, $\dot{\tilde{\mathbf{b}}}$, is retained.

Momenta moment:

$$\begin{aligned} \gamma &= \gamma_{CM} + \mathbf{b} \times \beta \\ &= \mathbf{J}_{CM} \omega + \mathbf{b} \times m \dot{\mathbf{x}}_{CM} \end{aligned} \quad (8.20)$$

$$\dot{\gamma}_{CM} = \boldsymbol{\omega} \times \mathbf{J}_{CM}\boldsymbol{\omega} + \mathbf{J}_{CM}\dot{\boldsymbol{\omega}} + (\mathbf{J}_{CM \text{ variable geometry}/t} + \mathbf{J}_{CM \text{ variable mass}/t}) \boldsymbol{\omega} \quad (8.21)$$

$$\begin{aligned} \dot{\gamma} &= \dot{\gamma}_{CM} + \dot{\mathbf{b}} \times \boldsymbol{\beta} + \mathbf{b} \times \dot{\boldsymbol{\beta}} \\ &= \boldsymbol{\omega} \times \mathbf{J}_{CM}\boldsymbol{\omega} + \mathbf{J}_{CM}\dot{\boldsymbol{\omega}} + (\mathbf{J}_{CM \text{ variable geometry}/t} + \mathbf{J}_{CM \text{ variable mass}/t}) \boldsymbol{\omega} \\ &\quad + (\boldsymbol{\omega} \times \mathbf{b} + \mathbf{R}\dot{\mathbf{b}}) \times \boldsymbol{\beta} + \mathbf{b} \times (m\ddot{\mathbf{x}}_{CM} + \dot{m}\dot{\mathbf{x}}_{CM}) \end{aligned} \quad (8.22)$$

Inertia moment of the whole body

$$\begin{aligned} \mathbf{m}_{in} &= \dot{\gamma}_{CM} + \mathbf{b} \times \dot{\boldsymbol{\beta}} \\ &= \dot{\gamma} - \dot{\mathbf{b}} \times \boldsymbol{\beta} \\ &= \boldsymbol{\omega} \times \mathbf{J}_{CM}\boldsymbol{\omega} + \mathbf{J}_{CM}\dot{\boldsymbol{\omega}} + (\mathbf{J}_{CM \text{ variable geometry}/t} + \mathbf{J}_{CM \text{ variable mass}/t}) \boldsymbol{\omega} \\ &\quad + \mathbf{b} \times (m\ddot{\mathbf{x}}_{CM} + \dot{m}\dot{\mathbf{x}}_{CM}) \end{aligned} \quad (8.23)$$

Inertia moment of the whole body minus the mass that is lost

$$\mathbf{m}_{in} = \dot{\gamma} - \dot{\mathbf{b}} \times \boldsymbol{\beta} - \mathbf{J}_{CM \text{ variable mass}/t} \boldsymbol{\omega} - \mathbf{b} \times \dot{m}\dot{\mathbf{x}}_{CM} \quad (8.24)$$

Only the portion of $\dot{\gamma}$ related to the change in mass is subtracted from the inertia forces; the portion related to the change in mass distribution, $\mathbf{J}_{CM \text{ variable geometry}/t}$, is retained.

Actually, in the constant mass code, the term $\dot{\mathbf{b}} \times \boldsymbol{\beta}$ is written as $-\mathbf{v} \times \boldsymbol{\beta}$, exploiting its structure:

$$\begin{aligned} \dot{\mathbf{b}} \times \boldsymbol{\beta} &= m(\boldsymbol{\omega} \times \mathbf{b}) \times (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{b}) \\ &= m(\boldsymbol{\omega} \times \mathbf{b}) \times \mathbf{v} \\ &= m(\mathbf{v} + \boldsymbol{\omega} \times \mathbf{b}) \times \mathbf{v} \\ &= \boldsymbol{\beta} \times \mathbf{v} \end{aligned} \quad (8.25)$$

In the variable mass case, this can be written as

$$\begin{aligned} \dot{\mathbf{b}} \times \boldsymbol{\beta} &= m(\boldsymbol{\omega} \times \mathbf{b} + \mathbf{R}\dot{\mathbf{b}}) \times (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{b} + \mathbf{R}\dot{\mathbf{b}}) \\ &= \boldsymbol{\beta} \times \mathbf{v} + m(\mathbf{R}\dot{\mathbf{b}}) \times (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{b} + \mathbf{R}\dot{\mathbf{b}}) + m(\boldsymbol{\omega} \times \mathbf{b}) \times \mathbf{R}\dot{\mathbf{b}} \\ &= \boldsymbol{\beta} \times \mathbf{v} + m(\mathbf{R}\dot{\mathbf{b}}) \times \mathbf{v} \end{aligned} \quad (8.26)$$

The correction term $\boldsymbol{\beta} \times \mathbf{v}$ is already applied to the equations of motion by the `AutomaticStructuralElem` element associated to the underlying node; the correction term $m(\mathbf{R}\dot{\mathbf{b}}) \times \mathbf{v}$ must be explicitly applied by the `DynamicVariableBody` element.

Linearization: currently the `DynamicVariableBody` element does not contribute to the Jacobian matrix with respect to the variable inertia properties; only the original contribution of the constant inertia properties element is provided. This implies that the residual is exact to the required tolerance, while the Jacobian matrix is not; convergence should not be compromised as soon as moderate inertia variations are used.

8.1.2 Static Structural Nodes

Static structural nodes represent a degeneration of the dynamic ones, when the inertia is structurally null. In that case, no momentum nor momenta moment definition equations are instantiated, and their derivatives are always null. As a consequence, static nodes only instantiate their equilibrium equations.

8.1.3 Dummy Nodes

Dummy nodes are special structural nodes that do not directly participate in the analysis. In fact, they do not provide degrees of freedom; they rather present information associated to other structural nodes in a different manner, which is output in the `.mov` file much like if they were regular nodes.

Offset Dummy Node. The `offset` dummy structural node outputs the configuration of a reference frame that is rigidly offset from a base node (subscript b) by \mathbf{b}_h and oriented by matrix \mathbf{R}_h according to the transformation

$$\mathbf{x} = \mathbf{x}_b + \mathbf{R}_b \mathbf{b}_h \quad (8.27a)$$

$$\mathbf{R} = \mathbf{R}_b \mathbf{R}_h \quad (8.27b)$$

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_b + \boldsymbol{\omega}_b \times \mathbf{R}_b \mathbf{b}_h \quad (8.27c)$$

$$\boldsymbol{\omega} = \boldsymbol{\omega}_b \quad (8.27d)$$

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}_b + (\dot{\boldsymbol{\omega}}_b \times + \boldsymbol{\omega}_b \times \boldsymbol{\omega}_b \times) \mathbf{R}_b \mathbf{b}_h \quad (8.27e)$$

$$\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}_b \quad (8.27f)$$

Reference Frame Dummy Node. The `reference frame` dummy structural node outputs the configuration of the base node (subscript b) in the reference frame of a reference node (subscript r), optionally offset by \mathbf{b}_h and with an orientation relative to r provided by matrix \mathbf{R}_h , according to the transformation

$$\bar{\mathbf{x}} = \mathbf{R}_h^T (\mathbf{R}_r^T (\mathbf{x}_b - \mathbf{x}_r) - \mathbf{b}_h) \quad (8.28a)$$

$$\bar{\mathbf{R}} = \mathbf{R}_h^T \mathbf{R}_r^T \mathbf{R}_b \quad (8.28b)$$

$$\bar{\dot{\mathbf{x}}} = \mathbf{R}_h^T \mathbf{R}_r^T (\dot{\mathbf{x}}_b - \dot{\mathbf{x}}_r - \boldsymbol{\omega}_r \times (\mathbf{x}_b - \mathbf{x}_r)) \quad (8.28c)$$

$$\bar{\boldsymbol{\omega}} = \mathbf{R}_h^T \mathbf{R}_r^T (\boldsymbol{\omega}_b - \boldsymbol{\omega}_r) \quad (8.28d)$$

$$\bar{\ddot{\mathbf{x}}} = \mathbf{R}_h^T \mathbf{R}_r^T (\ddot{\mathbf{x}}_b - \ddot{\mathbf{x}}_r - (\dot{\boldsymbol{\omega}}_r \times + \boldsymbol{\omega}_r \times \boldsymbol{\omega}_r \times) (\mathbf{x}_b - \mathbf{x}_r) - 2\boldsymbol{\omega}_r \times (\dot{\mathbf{x}}_b - \dot{\mathbf{x}}_r)) \quad (8.28e)$$

$$\bar{\dot{\boldsymbol{\omega}}} = \mathbf{R}_h^T \mathbf{R}_r^T (\dot{\boldsymbol{\omega}}_b - \dot{\boldsymbol{\omega}}_r - \boldsymbol{\omega}_r \times \boldsymbol{\omega}_b) \quad (8.28f)$$

Pivot Reference Frame Dummy Node. The `pivot reference frame` dummy node is a variant of the `reference frame` dummy node that outputs the configuration of the base node, expressed in the reference frame of the reference node, as if it were attached to the pivot node (subscript p), optionally offset by \mathbf{b}_k and with an orientation relative to p provided by matrix \mathbf{R}_k , according to the transformation

$$\hat{\mathbf{x}} = \mathbf{x}_p + \mathbf{R}_p (\mathbf{R}_k \bar{\mathbf{x}} + \mathbf{b}_k) \quad (8.29a)$$

$$\hat{\mathbf{R}} = \mathbf{R}_p \mathbf{R}_k \bar{\mathbf{R}} \quad (8.29b)$$

$$\hat{\dot{\mathbf{x}}} = \dot{\mathbf{x}}_p + \boldsymbol{\omega}_p \times \mathbf{R}_p (\mathbf{R}_k \bar{\mathbf{x}} + \mathbf{b}_k) + \mathbf{R}_p \mathbf{R}_k \dot{\bar{\mathbf{x}}} \quad (8.29c)$$

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_p + \mathbf{R}_p \mathbf{R}_k \bar{\boldsymbol{\omega}} \quad (8.29d)$$

$$\hat{\ddot{\mathbf{x}}} = \ddot{\mathbf{x}}_p + (\dot{\boldsymbol{\omega}}_p \times + \boldsymbol{\omega}_p \times \boldsymbol{\omega}_p \times) \mathbf{R}_p (\mathbf{R}_k \bar{\mathbf{x}} + \mathbf{b}_k) + 2\boldsymbol{\omega}_p \times \mathbf{R}_p \mathbf{R}_k \dot{\bar{\mathbf{x}}} + \mathbf{R}_p \mathbf{R}_k \ddot{\bar{\mathbf{x}}} \quad (8.29e)$$

$$\hat{\dot{\boldsymbol{\omega}}} = \dot{\boldsymbol{\omega}}_p + \boldsymbol{\omega}_p \times \mathbf{R}_p \mathbf{R}_k \bar{\boldsymbol{\omega}} + \mathbf{R}_p \mathbf{R}_k \dot{\bar{\boldsymbol{\omega}}} \quad (8.29f)$$

8.1.4 Relative Motion

MBDyn uses the absolute coordinates of the nodes, and their absolute orientation, to define the motion. Whenever the representation of the motion in a relative reference frame is required or convenient, it needs to be computed from the absolute one that is computed inside the code.

This is possible either run-time, by adding `dummy` structural nodes, or off-line, as a post-processing. For this purpose, the script `abs2rel.awk` is provided to convert the contents of the `.mov` file into the corresponding motion relative to a given node.

Absolute motion as function of relative (tilde) and reference (0) motion:

$$\mathbf{R} = \mathbf{R}_0 \tilde{\mathbf{R}} \quad (8.30a)$$

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{R}_0 \tilde{\mathbf{x}} \quad (8.30b)$$

$$\boldsymbol{\omega} = \boldsymbol{\omega}_0 + \mathbf{R}_0 \tilde{\boldsymbol{\omega}} \quad (8.30c)$$

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}} + \mathbf{R}_0 \dot{\tilde{\mathbf{x}}} \quad (8.30d)$$

$$\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}_0 + \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\boldsymbol{\omega}} + \mathbf{R}_0 \dot{\tilde{\boldsymbol{\omega}}} \quad (8.30e)$$

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}_0 + \dot{\boldsymbol{\omega}}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}} + \boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}} + 2\boldsymbol{\omega}_0 \times \mathbf{R}_0 \dot{\tilde{\mathbf{x}}} + \mathbf{R}_0 \ddot{\tilde{\mathbf{x}}} \quad (8.30f)$$

Relative motion (tilde) as function of absolute and reference (0) motion:

$$\tilde{\mathbf{R}} = \mathbf{R}_0^T \mathbf{R} \quad (8.31a)$$

$$\tilde{\mathbf{x}} = \mathbf{R}_0^T (\mathbf{x} - \mathbf{x}_0) \quad (8.31b)$$

$$\tilde{\boldsymbol{\omega}} = \mathbf{R}_0^T (\boldsymbol{\omega} - \boldsymbol{\omega}_0) \quad (8.31c)$$

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= \mathbf{R}_0^T (\dot{\mathbf{x}} - \dot{\mathbf{x}}_0 - \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}}) \\ &= \mathbf{R}_0^T (\dot{\mathbf{x}} - \dot{\mathbf{x}}_0) - (\mathbf{R}_0^T \boldsymbol{\omega}_0) \times \tilde{\mathbf{x}} \end{aligned} \quad (8.31d)$$

$$\begin{aligned} \dot{\tilde{\boldsymbol{\omega}}} &= \mathbf{R}_0^T (\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_0 - \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\boldsymbol{\omega}}) \\ &= \mathbf{R}_0^T (\dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_0) - (\mathbf{R}_0^T \boldsymbol{\omega}_0) \times \tilde{\boldsymbol{\omega}} \end{aligned} \quad (8.31e)$$

$$\begin{aligned} \ddot{\tilde{\mathbf{x}}} &= \mathbf{R}_0^T (\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_0 - \dot{\boldsymbol{\omega}}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}} - \boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}} - 2\boldsymbol{\omega}_0 \times \mathbf{R}_0 \dot{\tilde{\mathbf{x}}}) \\ &= \mathbf{R}_0^T (\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_0) - (\mathbf{R}_0^T \dot{\boldsymbol{\omega}}_0) \times \tilde{\mathbf{x}} \\ &\quad - (\mathbf{R}_0^T \boldsymbol{\omega}_0) \times (\mathbf{R}_0^T \boldsymbol{\omega}_0) \times \tilde{\mathbf{x}} - 2(\mathbf{R}_0^T \boldsymbol{\omega}_0) \times \dot{\tilde{\mathbf{x}}} \end{aligned} \quad (8.31f)$$

Perturbation, assuming reference motion is imposed:

$$\theta_\delta = \mathbf{R}_0 \tilde{\theta}_\delta \quad (8.32a)$$

$$\delta \mathbf{x} = \mathbf{R}_0 \delta \tilde{\mathbf{x}} \quad (8.32b)$$

$$\delta \boldsymbol{\omega} = \mathbf{R}_0 \delta \tilde{\boldsymbol{\omega}} \quad (8.32c)$$

$$\delta \dot{\mathbf{x}} = \boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \tilde{\mathbf{x}} + \mathbf{R}_0 \delta \dot{\tilde{\mathbf{x}}} \quad (8.32d)$$

$$\delta \dot{\boldsymbol{\omega}} = \boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \tilde{\boldsymbol{\omega}} + \mathbf{R}_0 \delta \dot{\tilde{\boldsymbol{\omega}}} \quad (8.32e)$$

$$\begin{aligned} \delta \ddot{\mathbf{x}} &= \dot{\boldsymbol{\omega}}_0 \times \mathbf{R}_0 \delta \tilde{\mathbf{x}} + \boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \tilde{\mathbf{x}} + 2\boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \dot{\tilde{\mathbf{x}}} + \mathbf{R}_0 \delta \ddot{\tilde{\mathbf{x}}} \\ &= (\dot{\boldsymbol{\omega}}_0 \times + \boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times) \mathbf{R}_0 \delta \tilde{\mathbf{x}} + 2\boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \dot{\tilde{\mathbf{x}}} + \mathbf{R}_0 \delta \ddot{\tilde{\mathbf{x}}} \end{aligned} \quad (8.32f)$$

Perturbation, when reference motion is independent:

$$\boldsymbol{\theta}_\delta = \boldsymbol{\theta}_{0\delta} + \mathbf{R}_0 \tilde{\boldsymbol{\theta}}_\delta \quad (8.33a)$$

$$\begin{aligned} \delta \mathbf{x} &= \delta \mathbf{x}_0 + \boldsymbol{\theta}_{0\delta} \times \mathbf{R}_0 \tilde{\mathbf{x}} + \mathbf{R}_0 \delta \tilde{\mathbf{x}} \\ &= \delta \mathbf{x}_0 - (\mathbf{R}_0 \tilde{\mathbf{x}}) \times \boldsymbol{\theta}_{0\delta} + \mathbf{R}_0 \delta \tilde{\mathbf{x}} \end{aligned} \quad (8.33b)$$

$$\begin{aligned} \delta \boldsymbol{\omega} &= \delta \boldsymbol{\omega}_0 + \boldsymbol{\theta}_{0\delta} \times \mathbf{R}_0 \tilde{\boldsymbol{\omega}} + \mathbf{R}_0 \delta \tilde{\boldsymbol{\omega}} \\ &= \delta \boldsymbol{\omega}_0 - (\mathbf{R}_0 \tilde{\boldsymbol{\omega}}) \times \boldsymbol{\theta}_{0\delta} + \mathbf{R}_0 \delta \tilde{\boldsymbol{\omega}} \end{aligned} \quad (8.33c)$$

$$\begin{aligned} \delta \dot{\mathbf{x}} &= \delta \dot{\mathbf{x}}_0 + \delta \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}} + \boldsymbol{\omega}_0 \times \boldsymbol{\theta}_{0\delta} \times \mathbf{R}_0 \tilde{\mathbf{x}} + \boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \tilde{\mathbf{x}} \\ &\quad + \boldsymbol{\theta}_{0\delta} \times \mathbf{R}_0 \dot{\tilde{\mathbf{x}}} + \mathbf{R}_0 \delta \dot{\tilde{\mathbf{x}}} \\ &= \delta \dot{\mathbf{x}}_0 - (\mathbf{R}_0 \tilde{\mathbf{x}}) \times \delta \boldsymbol{\omega}_0 - \left(\boldsymbol{\omega}_0 \times (\mathbf{R}_0 \tilde{\mathbf{x}}) \times + (\mathbf{R}_0 \dot{\tilde{\mathbf{x}}}) \times \right) \boldsymbol{\theta}_{0\delta} \\ &\quad + \boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \tilde{\mathbf{x}} + \mathbf{R}_0 \delta \dot{\tilde{\mathbf{x}}} \end{aligned} \quad (8.33d)$$

$$\begin{aligned} \delta \dot{\boldsymbol{\omega}} &= \delta \dot{\boldsymbol{\omega}}_0 + \delta \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\boldsymbol{\omega}} + \boldsymbol{\omega}_0 \times \boldsymbol{\theta}_{0\delta} \times \mathbf{R}_0 \tilde{\boldsymbol{\omega}} + \boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \tilde{\boldsymbol{\omega}} \\ &\quad + \boldsymbol{\theta}_{0\delta} \times \mathbf{R}_0 \dot{\tilde{\boldsymbol{\omega}}} + \mathbf{R}_0 \delta \dot{\tilde{\boldsymbol{\omega}}} \\ &= \delta \dot{\boldsymbol{\omega}}_0 - (\mathbf{R}_0 \tilde{\boldsymbol{\omega}}) \times \delta \boldsymbol{\omega}_0 - \left(\boldsymbol{\omega}_0 \times (\mathbf{R}_0 \tilde{\boldsymbol{\omega}}) \times + (\mathbf{R}_0 \dot{\tilde{\boldsymbol{\omega}}}) \times \right) \boldsymbol{\theta}_{0\delta} \\ &\quad + \boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \tilde{\boldsymbol{\omega}} + \mathbf{R}_0 \delta \dot{\tilde{\boldsymbol{\omega}}} \end{aligned} \quad (8.33e)$$

$$\begin{aligned} \delta \ddot{\mathbf{x}} &= \delta \ddot{\mathbf{x}}_0 \\ &\quad + \delta \dot{\boldsymbol{\omega}}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}} + \boldsymbol{\omega}_0 \times \boldsymbol{\theta}_{0\delta} \times \mathbf{R}_0 \tilde{\mathbf{x}} + \boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \tilde{\mathbf{x}} \\ &\quad + \delta \boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}} + \boldsymbol{\omega}_0 \times \delta \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}} \\ &\quad + \boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times \boldsymbol{\theta}_{0\delta} \times \mathbf{R}_0 \tilde{\mathbf{x}} + \boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \tilde{\mathbf{x}} \\ &\quad + 2\delta \boldsymbol{\omega}_0 \times \mathbf{R}_0 \dot{\tilde{\mathbf{x}}} + 2\boldsymbol{\omega}_0 \times \boldsymbol{\theta}_{0\delta} \times \mathbf{R}_0 \dot{\tilde{\mathbf{x}}} + 2\boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \dot{\tilde{\mathbf{x}}} \\ &\quad + \boldsymbol{\theta}_{0\delta} \times \mathbf{R}_0 \ddot{\tilde{\mathbf{x}}} + \mathbf{R}_0 \delta \ddot{\tilde{\mathbf{x}}} \\ &= \delta \ddot{\mathbf{x}}_0 - (\mathbf{R}_0 \tilde{\mathbf{x}}) \times \delta \dot{\boldsymbol{\omega}}_0 \\ &\quad - \left((\boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}}) \times + \boldsymbol{\omega}_0 \times (\mathbf{R}_0 \tilde{\mathbf{x}}) \times + 2(\mathbf{R}_0 \dot{\tilde{\mathbf{x}}}) \times \right) \delta \boldsymbol{\omega}_0 \\ &\quad - \left((\dot{\boldsymbol{\omega}}_0 \times + \boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times) (\mathbf{R}_0 \tilde{\mathbf{x}}) \times + 2\boldsymbol{\omega}_0 \times (\mathbf{R}_0 \dot{\tilde{\mathbf{x}}}) \times + (\mathbf{R}_0 \ddot{\tilde{\mathbf{x}}}) \times \right) \boldsymbol{\theta}_{0\delta} \\ &\quad + (\dot{\boldsymbol{\omega}}_0 \times + \boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times) \mathbf{R}_0 \delta \tilde{\mathbf{x}} + 2\boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \dot{\tilde{\mathbf{x}}} + \mathbf{R}_0 \delta \ddot{\tilde{\mathbf{x}}} \end{aligned} \quad (8.33f)$$

8.1.5 Motion Expressed in a Relative Reference Frame

An alternative, useful case is given by representing the absolute motion in a relative reference frame:

$$\tilde{\mathbf{R}} = \mathbf{R}_0^T \mathbf{R} \quad (8.34a)$$

$$\tilde{\mathbf{x}} = \mathbf{R}_0^T (\mathbf{x} - \mathbf{x}_0) \quad (8.34b)$$

$$\tilde{\boldsymbol{\omega}} = \mathbf{R}_0^T \boldsymbol{\omega} \quad (8.34c)$$

$$\dot{\tilde{\mathbf{x}}} = \mathbf{R}_0^T \dot{\mathbf{x}} \quad (8.34d)$$

$$\dot{\tilde{\boldsymbol{\omega}}} = \mathbf{R}_0^T \dot{\boldsymbol{\omega}} \quad (8.34e)$$

$$\ddot{\tilde{\mathbf{x}}} = \mathbf{R}_0^T \ddot{\mathbf{x}} \quad (8.34f)$$

8.1.6 Dynamics in a Relative Reference Frame

The rationale is that MBDyn is based on the assumption that the kinematics of the nodes and their equations of motion are formulated in the global reference frame. However, in some cases one may want to express the motion of the nodes in a relative reference frame.

The following discussion is intended to modify the inertial contribution to the equilibrium in order to reformulate the dynamical problem in a relative reference frame, whose motion is imposed. The constraints and the deformable components are unaffected, as they already intrinsically based on relative kinematics.

Momentum and momenta moments

$$\beta = m\dot{x}_0 + \omega_0 \times s_0 + R_0 \tilde{\beta} \quad (8.35a)$$

$$\gamma = s \times (\dot{x}_0 + \omega_0 \times R_0 \tilde{x}) + J\omega_0 + R_0 \tilde{\gamma} \quad (8.35b)$$

where

$$s_0 = R_0 \tilde{s}_0 \quad (8.36)$$

$$\tilde{s}_0 = \tilde{s} + m\tilde{x} \quad (8.37)$$

$$s = R_0 \tilde{s} \quad (8.38)$$

$$J = R_0 \tilde{J} R_0^T \quad (8.39)$$

Momentum and momenta moments derivative

$$\dot{\beta} = m\ddot{x}_0 + (\dot{\omega}_0 \times + \omega_0 \times \omega_0 \times) R_0 \tilde{s}_0 + 2\omega_0 \times R_0 \dot{\tilde{\beta}} + R_0 \dot{\tilde{\beta}} \quad (8.40a)$$

$$\begin{aligned} \dot{\gamma} &= \dot{s} \times (\dot{x}_0 + \omega_0 \times R_0 \tilde{x}) \\ &+ s \times (\ddot{x}_0 + (\dot{\omega}_0 \times + \omega_0 \times \omega_0 \times) R_0 \tilde{x} + \omega_0 \times R_0 \dot{\tilde{x}}) \\ &+ \omega_0 \times J\omega_0 + J\dot{\omega}_0 + R_0 (\tilde{\omega} \times \tilde{J} - \tilde{J}\tilde{\omega} \times) R_0^T \omega_0 \\ &+ \omega_0 \times R_0 \dot{\tilde{\gamma}} + R_0 \dot{\tilde{\gamma}} \end{aligned} \quad (8.40b)$$

Inertia forces and moments projected in the relative frame

$$\begin{aligned} \bar{f}_{in} &= R_0^T \dot{\beta} \\ &= m\ddot{x}_0 + (\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \tilde{s}_0 + 2\bar{\omega}_0 \times \tilde{\beta} + \underbrace{\dot{\tilde{\beta}}}_{\text{relative momentum}} \end{aligned} \quad (8.41a)$$

$$\begin{aligned} \bar{m}_{in} &= R_0^T (\dot{\gamma} + \dot{x} \times \beta) \\ &= \tilde{s} \times (\ddot{\bar{x}}_0 + (\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \tilde{x} + \bar{\omega}_0 \times \dot{\tilde{x}}) \\ &+ \bar{\omega}_0 \times \tilde{J} \bar{\omega}_0 + \tilde{J} \dot{\bar{\omega}}_0 + (\tilde{\omega} \times \tilde{J} - \tilde{J} \tilde{\omega} \times) \bar{\omega}_0 \\ &+ \dot{\tilde{x}} \times \bar{\omega}_0 \times \tilde{s} + \bar{\omega}_0 \times \dot{\tilde{\gamma}} + \underbrace{\dot{\tilde{x}} \times \dot{\tilde{\beta}} + \dot{\tilde{\gamma}}}_{\text{relative momenta moment}} \end{aligned} \quad (8.41b)$$

The contributions

$$\bar{f}_{in}^* = 2\bar{\omega}_0 \times \tilde{\beta} + \dot{\tilde{\beta}} \quad (8.42a)$$

$$\bar{m}_{in}^* = \bar{\omega}_0 \times \dot{\tilde{\gamma}} + \dot{\tilde{x}} \times \dot{\tilde{\beta}} + \dot{\tilde{\gamma}} \quad (8.42b)$$

belong to the **automatic structural** element associated to each dynamic node. As such, they are computed once for all irrespective of the number of rigid bodies that are connected to the node. The remaining contributions are computed by each rigid body.

Perturbation, assuming reference motion is imposed:

$$\delta \bar{\mathbf{f}}_{\text{in}} = (\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \left(m\delta \tilde{\mathbf{x}} - \tilde{\mathbf{s}} \times \tilde{\boldsymbol{\theta}}_\delta \right) \quad (8.43a)$$

$$+ 2\bar{\omega}_0 \times \delta \tilde{\boldsymbol{\beta}} + \delta \dot{\tilde{\boldsymbol{\beta}}} \quad (8.43b)$$

$$\begin{aligned} \delta \bar{\mathbf{m}}_{\text{in}} = & \left((\ddot{\bar{\mathbf{x}}}_0 + (\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \tilde{\mathbf{x}} + \bar{\omega}_0 \times \dot{\tilde{\mathbf{x}}}) \times \tilde{\mathbf{s}} \times \right. \\ & + (\bar{\omega}_0 + \tilde{\omega}) \times (\tilde{\mathbf{J}} \bar{\omega}_0 \times - (\tilde{\mathbf{J}} \bar{\omega}_0) \times) \\ & + \tilde{\mathbf{J}} (\dot{\bar{\omega}}_0 + \bar{\omega}_0 \times \tilde{\omega}) \times - (\tilde{\mathbf{J}} (\dot{\bar{\omega}}_0 + \bar{\omega}_0 \times \tilde{\omega})) \times \\ & \left. - \dot{\tilde{\mathbf{x}}} \times \bar{\omega}_0 \times \tilde{\mathbf{s}} \times \right) \tilde{\boldsymbol{\theta}}_\delta \\ & + (\tilde{\mathbf{J}} \bar{\omega}_0 \times - (\tilde{\mathbf{J}} \bar{\omega}_0) \times) \delta \tilde{\omega} \\ & + \tilde{\mathbf{s}} \times (\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \delta \tilde{\mathbf{x}} \\ & + (\tilde{\mathbf{s}} \times \bar{\omega}_0 \times - (\bar{\omega}_0 \times \tilde{\mathbf{s}}) \times - \tilde{\boldsymbol{\beta}} \times) \delta \dot{\tilde{\mathbf{x}}} \\ & + \bar{\omega}_0 \times \delta \tilde{\gamma} + \dot{\tilde{\mathbf{x}}} \times \delta \tilde{\boldsymbol{\beta}} + \delta \dot{\tilde{\gamma}} \end{aligned} \quad (8.43c)$$

To summarize:

$$\begin{aligned} \left\{ \begin{array}{l} \delta \bar{\mathbf{f}}_{\text{in}} \\ \delta \bar{\mathbf{m}}_{\text{in}} \end{array} \right\} = & \left[\begin{array}{cc} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{array} \right] \left\{ \begin{array}{l} \delta \dot{\tilde{\boldsymbol{\beta}}} \\ \delta \dot{\tilde{\gamma}} \end{array} \right\} + \left[\begin{array}{cc} 2\bar{\omega}_0 \times & \mathbf{0} \\ \dot{\tilde{\mathbf{x}}} \times & \bar{\omega} \times \end{array} \right] \left\{ \begin{array}{l} \delta \tilde{\boldsymbol{\beta}} \\ \delta \tilde{\gamma} \end{array} \right\} \\ & + \left[\begin{array}{cc} \mathbf{0} & \mathbf{0} \\ (\tilde{\mathbf{s}} \times \bar{\omega}_0 \times - (\bar{\omega}_0 \times \tilde{\mathbf{s}}) \times - \tilde{\boldsymbol{\beta}} \times) & (\tilde{\mathbf{J}} \bar{\omega}_0 \times - (\tilde{\mathbf{J}} \bar{\omega}_0) \times) \end{array} \right] \left\{ \begin{array}{l} \delta \dot{\tilde{\mathbf{x}}} \\ \delta \tilde{\omega} \end{array} \right\} \\ & + \left[\begin{array}{c} m(\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \\ \tilde{\mathbf{s}} \times (\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \end{array} \right] \delta \tilde{\mathbf{x}} \\ & + \left[\begin{array}{c} -(\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \tilde{\mathbf{s}} \times \\ \left(\begin{array}{c} (\ddot{\bar{\mathbf{x}}}_0 + (\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \tilde{\mathbf{x}} + \bar{\omega}_0 \times \dot{\tilde{\mathbf{x}}}) \times \tilde{\mathbf{s}} \times \\ - \dot{\tilde{\mathbf{x}}} \times \bar{\omega}_0 \times \tilde{\mathbf{s}} \times + \bar{\omega} \times (\tilde{\mathbf{J}} \bar{\omega}_0 \times - (\tilde{\mathbf{J}} \bar{\omega}_0) \times) \\ + \tilde{\mathbf{J}} (\dot{\bar{\omega}}_0 + \bar{\omega}_0 \times \tilde{\omega}) \times - (\tilde{\mathbf{J}} (\dot{\bar{\omega}}_0 + \bar{\omega}_0 \times \tilde{\omega})) \times \end{array} \right) \end{array} \right] \tilde{\boldsymbol{\theta}}_\delta \end{aligned} \quad (8.44)$$

8.1.7 Airstream Velocity

Aerodynamic elements need the absolute velocity of specific points in order to evaluate the boundary conditions for the computation of the aerodynamic forces.

The velocity of the airstream at a given absolute location \mathbf{x} is

$$\mathbf{v}_{\text{as}} = \mathbf{v}_{\text{as}}(\mathbf{x}, t), \quad (8.45)$$

which results from the superimposition of a time-dependent airstream velocity and of a time- and space-dependent gust velocity,

$$\mathbf{v}_{\text{as}} = \mathbf{v}_{\text{as}0}(t) + \mathbf{v}_{\text{g}}(\mathbf{x}, t). \quad (8.46)$$

The velocity of point \mathbf{x} with respect to the air at point \mathbf{x} itself is thus

$$\mathbf{v}_a = \dot{\mathbf{x}} - \mathbf{v}_{as}. \quad (8.47)$$

When the motion of the point is expressed in the relative reference frame, the position of the point is expressed by Eq. (8.30d). The velocity of Eq. (8.47) becomes

$$\mathbf{v}_a = \dot{\mathbf{x}}_0 + \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}} + \mathbf{R}_0 \dot{\tilde{\mathbf{x}}} - \mathbf{v}_{as}. \quad (8.48)$$

This velocity, projected in the relative reference frame, is

$$\begin{aligned} \bar{\mathbf{v}}_a &= \mathbf{R}_0^T \mathbf{v}_a \\ &= \dot{\tilde{\mathbf{x}}}_0 + \bar{\boldsymbol{\omega}}_0 \times \tilde{\mathbf{x}} + \dot{\tilde{\mathbf{x}}} - \bar{\mathbf{v}}_{as}. \end{aligned} \quad (8.49)$$

When referring aerodynamic elements to the relative frame, the equivalent airstream speed is thus

$$\mathbf{v}_{as\text{ equivalent}} = \bar{\mathbf{v}}_{as} - \dot{\tilde{\mathbf{x}}}_0 - \bar{\boldsymbol{\omega}}_0 \times \tilde{\mathbf{x}}, \quad (8.50)$$

so that the velocity of the air at point $\tilde{\mathbf{x}}$ is computed as

$$\bar{\mathbf{v}}_a = \dot{\tilde{\mathbf{x}}} - \mathbf{v}_{as\text{ equivalent}}, \quad (8.51)$$

in analogy with Eq. (8.47). This allows to confine the relative reference frame modification of the aerodynamic forces in the computation of the airstream velocity.

8.1.8 Implementation Notes

Kinematics

The relative reference frame dynamics is under development. In the current design, the whole model is subjected to a single relative reference frame. The “ground” is intended to be rigidly connected to the relative reference frame (e.g. all “pin” joints and the “clamp” joint). Deformable elements and kinematic constraints do not discriminate between absolute and relative reference frame, since they only deal with the relative kinematics of the nodes they connect.

The relative frame motion is delegated to the `DataManager`. The `StructNode` is responsible for providing a pointer to a `RigidBodyKinematics` object that describes the relative frame motion.

The kinematic parameters received in input with the `rigid body kinematics` card in the `control` data section are intended expressed in the global reference frame. Note however that the calls to the `Get*()` methods of the `RigidBodyKinematics` class return them in the relative reference frame, namely pre-multiplied by the transpose of the orientation matrix:

$$\begin{aligned} \text{GetR(): } &\mathbf{R}_0 = \{ \exp(\text{abs_orientation_vector} \times) \\ &\quad | \text{ abs_orientation } \} \end{aligned} \quad (8.52a)$$

$$\text{GetX(): } \bar{\mathbf{x}}_0 = \mathbf{R}_0^T \text{abs_position} \quad (8.52b)$$

$$\text{GetW(): } \bar{\boldsymbol{\omega}}_0 = \mathbf{R}_0^T \text{abs_angular_velocity} \quad (8.52c)$$

$$\text{GetV(): } \dot{\bar{\mathbf{x}}}_0 = \mathbf{R}_0^T \text{abs_velocity} \quad (8.52d)$$

$$\text{GetWP(): } \dot{\bar{\boldsymbol{\omega}}}_0 = \mathbf{R}_0^T \text{abs_angular_acceleration} \quad (8.52e)$$

$$\text{GetXPP(): } \ddot{\bar{\mathbf{x}}}_0 = \mathbf{R}_0^T \text{abs_acceleration} \quad (8.52f)$$

Dynamics

The equations of motion (force and moment equilibrium) and the definitions of the momentum and momenta moment are not altered.

The `AutomaticStructElem` element takes care of the portion of relative frame inertia forces that depends on the momentum and on the momenta moment. The `DynamicBody` element takes care of the remaining portion of relative frame inertia forces. The `StaticBody` element has been modified accordingly.

Air Properties

The `AirProperties` element takes care of the relative frame motion in order to project the absolute reference frame airstream velocity into the relative reference frame, according to Section 8.1.7.

All other built-in aerodynamic elements should naturally inherit the correct air velocity at the reference points.

Future Development

Future development may allow to have multiple relative reference frame submodels, connected to each other and to the absolute reference frame submodel, if any, by dedicated algebraic constraints.

8.1.9 Pseudo-Velocities Approach

NOTE: this is not currently implemented, nor foreseen.

The pseudo-velocities approach consists in replacing the momentum and momenta moment definitions with the definitions of the angular and linear velocities,

$$\mathbf{v} = \dot{\mathbf{x}} \quad (8.53a)$$

$$\boldsymbol{\omega} = \text{ax}(\dot{\mathbf{R}}\mathbf{R}^T) \quad (8.53b)$$

An advantage is that this approach allows an easier implementation of constraint stabilization.

Kinematics:

$$\mathbf{R} := \text{node orientation matrix} \quad (8.54a)$$

$$\mathbf{x} := \text{node position} \quad (8.54b)$$

Angular and linear velocity, $\boldsymbol{\omega}$ and \mathbf{v} :

$$\boldsymbol{\omega} = \text{ax}(\dot{\mathbf{R}}\mathbf{R}^T) \quad (8.55a)$$

$$\mathbf{v} = \dot{\mathbf{x}} \quad (8.55b)$$

Inertia forces and moments:

$$\mathbf{f}_{\text{in}} = m\dot{\mathbf{v}}_{\text{CM}} \quad (8.56a)$$

$$\mathbf{m}_{\text{in}} = \boldsymbol{\omega} \times \mathbf{J}_{\text{CM}}\boldsymbol{\omega} + \mathbf{J}_{\text{CM}}\dot{\boldsymbol{\omega}} + \mathbf{s} \times \dot{\mathbf{v}}_{\text{CM}} \quad (8.56b)$$

where

$$\mathbf{s} \quad \text{static moment with respect to the node position} \quad (8.57\text{a})$$

$$\mathbf{J}_{\text{CM}} \quad \text{inertia tensor with respect to the center of mass} \quad (8.57\text{b})$$

$$\mathbf{x}_{\text{CM}} = \mathbf{x} + \frac{1}{m}\mathbf{s} \quad \text{position of the center of mass} \quad (8.57\text{c})$$

$$\mathbf{v}_{\text{CM}} = \mathbf{v} + \frac{1}{m}\boldsymbol{\omega} \times \mathbf{s} \quad \text{velocity of the center of mass} \quad (8.57\text{d})$$

$$\dot{\mathbf{v}}_{\text{CM}} = \dot{\mathbf{v}} + \frac{1}{m}(\dot{\boldsymbol{\omega}} \times + \boldsymbol{\omega} \times \boldsymbol{\omega} \times) \mathbf{s} \quad (8.57\text{e})$$

Perturbation

$$\delta\mathbf{s} = \boldsymbol{\theta}_\delta \times \mathbf{s} \quad (8.58\text{a})$$

$$\delta\mathbf{J}_{\text{CM}} = \boldsymbol{\theta}_\delta \times \mathbf{J}_{\text{CM}} - \mathbf{J}_{\text{CM}}\boldsymbol{\theta}_\delta \times \quad (8.58\text{b})$$

$$\begin{aligned} \delta\mathbf{x}_{\text{CM}} &= \delta\mathbf{x} + \frac{1}{m}\delta\mathbf{s} \\ &= \delta\mathbf{x} - \frac{1}{m}\mathbf{s} \times \boldsymbol{\theta}_\delta \end{aligned} \quad (8.58\text{c})$$

$$\begin{aligned} \mathbf{v}_{\text{CM}} &= \delta\mathbf{v} + \frac{1}{m}\delta\boldsymbol{\omega} \times \mathbf{s} + \frac{1}{m}\boldsymbol{\omega} \times \delta\mathbf{s} \\ &= \delta\mathbf{v} - \frac{1}{m}\mathbf{s} \times \delta\boldsymbol{\omega} - \frac{1}{m}\boldsymbol{\omega} \times \mathbf{s} \times \boldsymbol{\theta}_\delta \end{aligned} \quad (8.58\text{d})$$

$$\begin{aligned} \dot{\mathbf{v}}_{\text{CM}} &= \dot{\mathbf{v}} + \frac{1}{m}(\delta\dot{\boldsymbol{\omega}} \times \mathbf{s} + \delta\boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{s} + \boldsymbol{\omega} \times \delta\boldsymbol{\omega} \times \mathbf{s} + (\dot{\boldsymbol{\omega}} \times + \boldsymbol{\omega} \times \boldsymbol{\omega} \times) \delta\mathbf{s}) \\ &= \dot{\mathbf{v}} - \frac{1}{m}\mathbf{s} \times \delta\dot{\boldsymbol{\omega}} - \frac{1}{m}((\boldsymbol{\omega} \times \mathbf{s}) \times + \boldsymbol{\omega} \times \mathbf{s} \times) \delta\boldsymbol{\omega} \\ &\quad - \frac{1}{m}(\dot{\boldsymbol{\omega}} \times + \boldsymbol{\omega} \times \boldsymbol{\omega} \times) \mathbf{s} \times \boldsymbol{\theta}_\delta \end{aligned} \quad (8.58\text{e})$$

Inertia forces and moments with respect to nodal quantities:

$$\begin{aligned} \mathbf{f}_{\text{in}} &= m\dot{\mathbf{v}}_{\text{CM}} \\ &= m\dot{\mathbf{v}} + (\dot{\boldsymbol{\omega}} \times + \boldsymbol{\omega} \times \boldsymbol{\omega} \times) \mathbf{s} \end{aligned} \quad (8.59\text{a})$$

$$\begin{aligned} \mathbf{m}_{\text{in}} &= \boldsymbol{\omega} \times \mathbf{J}_{\text{CM}}\boldsymbol{\omega} + \mathbf{J}_{\text{CM}}\dot{\boldsymbol{\omega}} + \mathbf{s} \times \dot{\mathbf{v}}_{\text{CM}} \\ &= \mathbf{s} \times \dot{\mathbf{v}} + \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{aligned} \quad (8.59\text{b})$$

with

$$\mathbf{J} = \mathbf{J}_{\text{CM}} - \frac{1}{m}\mathbf{s} \times \mathbf{s} \times \quad (8.60)$$

the inertia tensor with respect to the node.

Perturbation:

$$\begin{aligned} \delta\mathbf{f}_{\text{in}} &= m\delta\dot{\mathbf{v}} - \mathbf{s} \times \delta\dot{\boldsymbol{\omega}} - ((\boldsymbol{\omega} \times \mathbf{s}) \times + \boldsymbol{\omega} \times \mathbf{s} \times) \delta\boldsymbol{\omega} \\ &\quad - (\dot{\boldsymbol{\omega}} \times + \boldsymbol{\omega} \times \boldsymbol{\omega} \times) \mathbf{s} \times \boldsymbol{\theta}_\delta \end{aligned} \quad (8.61\text{a})$$

$$\begin{aligned} \delta\mathbf{m}_{\text{in}} &= \mathbf{s} \times \delta\dot{\mathbf{v}} + \mathbf{J}\delta\dot{\boldsymbol{\omega}} + (\boldsymbol{\omega} \times \mathbf{J} - (\mathbf{J}\boldsymbol{\omega}) \times) \delta\boldsymbol{\omega} \\ &\quad + (\boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega}) \times - (\mathbf{J}\boldsymbol{\omega}) \times) + \mathbf{J}\dot{\boldsymbol{\omega}} \times - (\mathbf{J}\dot{\boldsymbol{\omega}}) \times + \dot{\mathbf{v}} \times \mathbf{s} \times \boldsymbol{\theta}_\delta \end{aligned} \quad (8.61\text{b})$$

Relative frame kinematics:

$$\mathbf{R} = \mathbf{R}_0 \tilde{\mathbf{R}} \quad (8.62\text{a})$$

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{R}_0 \tilde{\mathbf{x}} \quad (8.62\text{b})$$

$$\boldsymbol{\omega} = \boldsymbol{\omega}_0 + \mathbf{R}_0 \tilde{\boldsymbol{\omega}} \quad (8.62\text{c})$$

$$\mathbf{v} = \mathbf{v}_0 + \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\mathbf{x}} + \mathbf{R}_0 \tilde{\mathbf{v}} \quad (8.62\text{d})$$

$$\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}_0 + \boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\boldsymbol{\omega}} + \mathbf{R}_0 \dot{\tilde{\boldsymbol{\omega}}} \quad (8.62\text{e})$$

$$\dot{\mathbf{v}} = \dot{\mathbf{v}}_0 + (\dot{\boldsymbol{\omega}}_0 \times + \boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times) \mathbf{R}_0 \tilde{\mathbf{x}} + 2\boldsymbol{\omega}_0 \times \mathbf{R}_0 \tilde{\mathbf{v}} + \mathbf{R}_0 \dot{\tilde{\mathbf{v}}} \quad (8.62\text{f})$$

Perturbation:

$$\theta_\delta = \mathbf{R}_0 \tilde{\theta}_\delta \quad (8.63\text{a})$$

$$\delta \mathbf{x} = \mathbf{R}_0 \delta \tilde{\mathbf{x}} \quad (8.63\text{b})$$

$$\delta \boldsymbol{\omega} = \mathbf{R}_0 \delta \tilde{\boldsymbol{\omega}} \quad (8.63\text{c})$$

$$\delta \mathbf{v} = \boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \tilde{\mathbf{x}} + \mathbf{R}_0 \delta \tilde{\mathbf{v}} \quad (8.63\text{d})$$

$$\delta \dot{\boldsymbol{\omega}} = \boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \tilde{\boldsymbol{\omega}} + \mathbf{R}_0 \delta \dot{\tilde{\boldsymbol{\omega}}} \quad (8.63\text{e})$$

$$\delta \dot{\mathbf{v}} = (\dot{\boldsymbol{\omega}}_0 \times + \boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times) \mathbf{R}_0 \delta \tilde{\mathbf{x}} + 2\boldsymbol{\omega}_0 \times \mathbf{R}_0 \delta \tilde{\mathbf{v}} + \mathbf{R}_0 \delta \dot{\tilde{\mathbf{v}}} \quad (8.63\text{f})$$

Relative frame kinematics projected in the relative frame:

$$\bar{\mathbf{R}} = \tilde{\mathbf{R}} \quad (8.64\text{a})$$

$$\bar{\mathbf{x}} = \mathbf{R}_0^T \mathbf{x}_0 + \tilde{\mathbf{x}} \quad (8.64\text{b})$$

$$\bar{\boldsymbol{\omega}} = \mathbf{R}_0^T \boldsymbol{\omega}_0 + \tilde{\boldsymbol{\omega}} \quad (8.64\text{c})$$

$$\bar{\mathbf{v}} = \mathbf{R}_0^T \mathbf{v}_0 + \bar{\boldsymbol{\omega}}_0 \times \tilde{\mathbf{x}} + \tilde{\mathbf{v}} \quad (8.64\text{d})$$

$$\dot{\bar{\boldsymbol{\omega}}} = \mathbf{R}_0^T \dot{\boldsymbol{\omega}}_0 + \bar{\boldsymbol{\omega}}_0 \times \tilde{\boldsymbol{\omega}} + \dot{\tilde{\boldsymbol{\omega}}} \quad (8.64\text{e})$$

$$\dot{\bar{\mathbf{v}}} = \mathbf{R}_0^T \dot{\mathbf{v}}_0 + (\dot{\bar{\boldsymbol{\omega}}}_0 \times + \bar{\boldsymbol{\omega}}_0 \times \bar{\boldsymbol{\omega}}_0 \times) \tilde{\mathbf{x}} + 2\bar{\boldsymbol{\omega}}_0 \times \tilde{\mathbf{v}} + \dot{\tilde{\mathbf{v}}} \quad (8.64\text{f})$$

Perturbation projected in the relative frame:

$$\bar{\theta}_\delta = \tilde{\theta}_\delta \quad (8.65\text{a})$$

$$\delta \bar{\mathbf{x}} = \delta \tilde{\mathbf{x}} \quad (8.65\text{b})$$

$$\delta \bar{\boldsymbol{\omega}} = \delta \tilde{\boldsymbol{\omega}} \quad (8.65\text{c})$$

$$\delta \bar{\mathbf{v}} = \bar{\boldsymbol{\omega}}_0 \times \delta \tilde{\mathbf{x}} + \delta \tilde{\mathbf{v}} \quad (8.65\text{d})$$

$$\dot{\delta \bar{\boldsymbol{\omega}}} = \bar{\boldsymbol{\omega}}_0 \times \delta \tilde{\boldsymbol{\omega}} + \dot{\delta \tilde{\boldsymbol{\omega}}} \quad (8.65\text{e})$$

$$\dot{\delta \bar{\mathbf{v}}} = (\dot{\bar{\boldsymbol{\omega}}}_0 \times + \bar{\boldsymbol{\omega}}_0 \times \bar{\boldsymbol{\omega}}_0 \times) \delta \tilde{\mathbf{x}} + 2\bar{\boldsymbol{\omega}}_0 \times \delta \tilde{\mathbf{v}} + \dot{\delta \tilde{\mathbf{v}}} \quad (8.65\text{f})$$

Inertia forces and moments with respect to nodal quantities, in a relative frame:

$$\bar{\mathbf{f}}_{\text{in}} = m \dot{\tilde{\mathbf{v}}}_{\text{CM}} \quad (8.66\text{a})$$

$$\bar{\mathbf{m}}_{\text{in}} = \tilde{\mathbf{s}} \times \dot{\tilde{\mathbf{v}}}_{\text{CM}} + \bar{\boldsymbol{\omega}} \times \tilde{\mathbf{J}}_{\text{CM}} \bar{\boldsymbol{\omega}} + \tilde{\mathbf{J}}_{\text{CM}} \dot{\bar{\boldsymbol{\omega}}} \quad (8.66\text{b})$$

Perturbation

$$\begin{aligned}\delta \bar{\mathbf{f}}_{\text{in}} &= m \delta \dot{\tilde{\mathbf{v}}}_{\text{CM}} \\ &= m \delta \dot{\tilde{\mathbf{v}}} + 2m \bar{\omega}_0 \times \delta \tilde{\mathbf{v}} + m (\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \delta \tilde{\mathbf{x}} \\ &\quad - \tilde{\mathbf{s}} \times \delta \dot{\tilde{\omega}} - (\tilde{\mathbf{s}} \times \bar{\omega}_0 \times + \bar{\omega} \times \tilde{\mathbf{s}} \times - (\tilde{\mathbf{s}} \times \bar{\omega}) \times) \delta \tilde{\omega} \\ &\quad - (\dot{\bar{\omega}} \times + \bar{\omega} \times \bar{\omega} \times) \tilde{\mathbf{s}} \times \tilde{\theta}_\delta\end{aligned}\tag{8.67a}$$

$$\begin{aligned}\delta \bar{\mathbf{m}}_{\text{in}} &= \tilde{\mathbf{s}} \times \delta \dot{\tilde{\mathbf{v}}}_{\text{CM}} + \delta \tilde{\mathbf{s}} \times \dot{\tilde{\mathbf{v}}}_{\text{CM}} \\ &\quad + \tilde{\mathbf{J}}_{\text{CM}} \delta \dot{\bar{\omega}} + (\bar{\omega} \times \tilde{\mathbf{J}}_{\text{CM}} - (\tilde{\mathbf{J}}_{\text{CM}} \bar{\omega}) \times) \delta \bar{\omega} \\ &\quad + (\bar{\omega} \times (\tilde{\mathbf{J}}_{\text{CM}} \bar{\omega} \times - (\tilde{\mathbf{J}}_{\text{CM}} \bar{\omega}) \times) + \tilde{\mathbf{J}}_{\text{CM}} \dot{\bar{\omega}} \times - (\tilde{\mathbf{J}}_{\text{CM}} \dot{\bar{\omega}}) \times) \tilde{\theta}_\delta \\ &= \tilde{\mathbf{s}} \times \delta \dot{\tilde{\mathbf{v}}} \\ &\quad + 2\tilde{\mathbf{s}} \times \bar{\omega}_0 \times \delta \tilde{\mathbf{v}} \\ &\quad + \tilde{\mathbf{s}} \times (\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \delta \tilde{\mathbf{x}} \\ &\quad + \tilde{\mathbf{J}} \delta \dot{\tilde{\omega}} \\ &\quad + (\tilde{\mathbf{J}} \bar{\omega}_0 \times + \bar{\omega} \times \tilde{\mathbf{J}} - (\tilde{\mathbf{J}} \bar{\omega}) \times) \delta \tilde{\omega} \\ &\quad + (\tilde{\mathbf{J}} \dot{\bar{\omega}} \times - (\tilde{\mathbf{J}} \dot{\bar{\omega}}) \times + \bar{\omega} \times (\tilde{\mathbf{J}} \bar{\omega} \times - (\tilde{\mathbf{J}} \bar{\omega}) \times) + \dot{\tilde{\mathbf{v}}} \times \tilde{\mathbf{s}} \times) \tilde{\theta}_\delta\end{aligned}\tag{8.67b}$$

To summarize:

$$\begin{aligned}\left\{ \begin{array}{l} \delta \bar{\mathbf{f}}_{\text{in}} \\ \delta \bar{\mathbf{m}}_{\text{in}} \end{array} \right\} &= \left[\begin{array}{cc} m \mathbf{I} & -\tilde{\mathbf{s}} \times \\ \tilde{\mathbf{s}} \times & \tilde{\mathbf{J}} \end{array} \right] \left\{ \begin{array}{l} \delta \dot{\tilde{\mathbf{v}}} \\ \delta \dot{\tilde{\omega}} \end{array} \right\} \\ &+ \left[\begin{array}{cc} 2m \bar{\omega}_0 \times & -(\tilde{\mathbf{s}} \times \bar{\omega}_0 \times + \bar{\omega} \times \tilde{\mathbf{s}} \times - (\tilde{\mathbf{s}} \times \bar{\omega}) \times) \\ 2\tilde{\mathbf{s}} \times \bar{\omega}_0 \times & (\tilde{\mathbf{J}} \bar{\omega}_0 \times + \bar{\omega} \times \tilde{\mathbf{J}} - (\tilde{\mathbf{J}} \bar{\omega}) \times) \end{array} \right] \left\{ \begin{array}{l} \delta \tilde{\mathbf{v}} \\ \delta \tilde{\omega} \end{array} \right\} \\ &+ \left[\begin{array}{c} m (\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \\ \tilde{\mathbf{s}} \times (\dot{\bar{\omega}}_0 \times + \bar{\omega}_0 \times \bar{\omega}_0 \times) \end{array} \right] \delta \tilde{\mathbf{x}} \\ &+ \left[\begin{array}{c} -(\dot{\bar{\omega}} \times + \bar{\omega} \times \bar{\omega} \times) \tilde{\mathbf{s}} \times \\ (\tilde{\mathbf{J}} \dot{\bar{\omega}} \times - (\tilde{\mathbf{J}} \dot{\bar{\omega}}) \times + \bar{\omega} \times (\tilde{\mathbf{J}} \bar{\omega} \times - (\tilde{\mathbf{J}} \bar{\omega}) \times) + \dot{\tilde{\mathbf{v}}} \times \tilde{\mathbf{s}} \times) \end{array} \right] \tilde{\theta}_\delta\end{aligned}\tag{8.68}$$

Chapter 9

Constraints

9.1 Algebraic Constraints

Consider a holonomic constraint equation of the form

$$\phi(\mathbf{q}, t) = 0 \quad (9.1)$$

Its time derivative and perturbation yields

$$\begin{aligned} \delta \frac{d}{dt} (\boldsymbol{\mu}^T \phi) &= \delta \dot{\boldsymbol{\mu}}^T \phi + \delta \boldsymbol{\phi}^T \dot{\boldsymbol{\mu}} + \delta \boldsymbol{\mu}^T \dot{\boldsymbol{\phi}} + \delta \dot{\boldsymbol{\phi}}^T \boldsymbol{\mu} \\ &= \delta \dot{\boldsymbol{\mu}}^T \phi + \delta \mathbf{q}^T \boldsymbol{\phi}_{/\mathbf{q}}^T \dot{\boldsymbol{\mu}} + \delta \boldsymbol{\mu}^T (\boldsymbol{\phi}_{/\mathbf{q}} \dot{\mathbf{q}} + \boldsymbol{\phi}_{/t}) \\ &\quad + \left(\delta \dot{\mathbf{q}}^T \boldsymbol{\phi}_{/\mathbf{q}}^T + \delta \mathbf{q}^T (\boldsymbol{\phi}_{/\mathbf{q}} \dot{\mathbf{q}} + \boldsymbol{\phi}_{/t})_{/\mathbf{q}}^T \right) \boldsymbol{\mu} \\ &= \delta \dot{\mathbf{q}}^T \boldsymbol{\phi}_{/\mathbf{q}}^T \boldsymbol{\mu} \\ &\quad + \delta \mathbf{q}^T \left((\boldsymbol{\phi}_{/\mathbf{q}} \dot{\mathbf{q}} + \boldsymbol{\phi}_{/t})_{/\mathbf{q}}^T \boldsymbol{\mu} + \boldsymbol{\phi}_{/\mathbf{q}}^T \dot{\boldsymbol{\mu}} \right) \\ &\quad + \delta \dot{\boldsymbol{\mu}}^T \boldsymbol{\phi} \\ &\quad + \delta \boldsymbol{\mu}^T (\boldsymbol{\phi}_{/\mathbf{q}} \dot{\mathbf{q}} + \boldsymbol{\phi}_{/t}) \end{aligned} \quad (9.2)$$

Its linearization yields

$$\begin{aligned} \Delta \left(\delta \frac{d}{dt} (\boldsymbol{\mu}^T \phi) \right) &= \delta \dot{\mathbf{q}}^T \left(\boldsymbol{\phi}_{/\mathbf{q}}^T \Delta \boldsymbol{\mu} + (\boldsymbol{\phi}_{/\mathbf{q}}^T \boldsymbol{\mu})_{/\mathbf{q}} \Delta \mathbf{q} \right) \\ &\quad + \delta \mathbf{q}^T \left((\boldsymbol{\phi}_{/\mathbf{q}} \dot{\mathbf{q}} + \boldsymbol{\phi}_{/t})_{/\mathbf{q}}^T \Delta \boldsymbol{\mu} + \boldsymbol{\phi}_{/\mathbf{q}} \Delta \dot{\boldsymbol{\mu}} + (\boldsymbol{\phi}_{/\mathbf{q}} \Delta \dot{\mathbf{q}})_{/\mathbf{q}}^T \boldsymbol{\mu} \right. \\ &\quad \left. + \left((\boldsymbol{\phi}_{/\mathbf{q}} \dot{\mathbf{q}} + \boldsymbol{\phi}_{/t})_{/\mathbf{q}}^T \boldsymbol{\mu} + \boldsymbol{\phi}_{/\mathbf{q}}^T \dot{\boldsymbol{\mu}} \right)_{/\mathbf{q}} \Delta \mathbf{q} \right) \\ &\quad + \delta \dot{\boldsymbol{\mu}}^T \boldsymbol{\phi}_{/\mathbf{q}} \Delta \mathbf{q} \\ &\quad + \delta \boldsymbol{\mu}^T \left(\boldsymbol{\phi}_{/\mathbf{q}} \Delta \dot{\mathbf{q}} + (\boldsymbol{\phi}_{/\mathbf{q}} \dot{\mathbf{q}} + \boldsymbol{\phi}_{/t})_{/\mathbf{q}} \Delta \mathbf{q} \right) \end{aligned} \quad (9.3)$$

An unconstrained problem of the form

$$M \dot{\mathbf{q}} - \mathbf{p} = \mathbf{0} \quad (9.4a)$$

$$\dot{\mathbf{p}} = \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (9.4b)$$

becomes

$$\mathbf{M}\dot{\mathbf{q}} - \mathbf{p} + \boldsymbol{\phi}_{/\mathbf{q}}^T \boldsymbol{\mu} = \mathbf{0} \quad (9.5a)$$

$$\dot{\mathbf{p}} + (\boldsymbol{\phi}_{/\mathbf{q}}\dot{\mathbf{q}} + \boldsymbol{\phi}_{/t})_{/\mathbf{q}}^T \boldsymbol{\mu} + \boldsymbol{\phi}_{/\mathbf{q}}^T \dot{\boldsymbol{\mu}} = \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (9.5b)$$

$$\boldsymbol{\phi} = \mathbf{0} \quad (9.5c)$$

$$\boldsymbol{\phi}_{/\mathbf{q}} \mathbf{M}^{-1} \mathbf{p} + \boldsymbol{\phi}_{/t} = \mathbf{0}, \quad (9.5d)$$

where $\dot{\mathbf{q}} = \mathbf{M}^{-1} \mathbf{p}$ has been used in Eq. (9.5d). Actually, $\dot{\boldsymbol{\mu}}$ is independent of $\boldsymbol{\mu}$; $\boldsymbol{\mu}$ by definition is zero if the exact solution is considered. So $\dot{\boldsymbol{\mu}}$ is redefined as $\dot{\boldsymbol{\mu}} = \boldsymbol{\lambda}$:

$$\mathbf{M}\dot{\mathbf{q}} - \mathbf{p} + \boldsymbol{\phi}_{/\mathbf{q}}^T \boldsymbol{\mu} = \mathbf{0} \quad (9.6a)$$

$$\dot{\mathbf{p}} + (\boldsymbol{\phi}_{/\mathbf{q}}\dot{\mathbf{q}} + \boldsymbol{\phi}_{/t})_{/\mathbf{q}}^T \boldsymbol{\mu} + \boldsymbol{\phi}_{/\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (9.6b)$$

$$\boldsymbol{\phi} = \mathbf{0} \quad (9.6c)$$

$$\boldsymbol{\phi}_{/\mathbf{q}} \mathbf{M}^{-1} \mathbf{p} + \boldsymbol{\phi}_{/t} = \mathbf{0}. \quad (9.6d)$$

If $\boldsymbol{\mu} = 0$ then a conventional DAE of index 3 results, i.e.

$$\mathbf{M}\dot{\mathbf{q}} - \mathbf{p} = \mathbf{0} \quad (9.7a)$$

$$\dot{\mathbf{p}} + \boldsymbol{\phi}_{/\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (9.7b)$$

$$\boldsymbol{\phi} = \mathbf{0}, \quad (9.7c)$$

while the previous form is known as stabilized index 2 form. The latter is typically used throughout MBDyn, while the former is used in the initial assembly and, occasionally, in specific constraints.

9.1.1 Clamp Joint

The clamp joint (`Clamp`) imposes the position and the orientation of a node.

Files. It is implemented in files

`mbdyn/struct/genj.h`

`mbdyn/struct/genj.cc`

Definitions.

$$\mathbf{d}_c = \mathbf{x} - \mathbf{x}_0 \quad (9.8)$$

$$\boldsymbol{\theta}_c = \text{ax}(\exp^{-1}(\mathbf{R}\mathbf{R}_0^T)) \quad (9.9)$$

Constraint Equation.

$$\mathbf{d}_c = \mathbf{0} \quad (9.10)$$

$$\boldsymbol{\theta}_c = \mathbf{0} \quad (9.11)$$

Perturbation.

$$\delta \mathbf{d}_c = \delta \mathbf{x} \quad (9.12)$$

$$\delta \boldsymbol{\theta}_c = \boldsymbol{\Gamma}^{-1}(\boldsymbol{\theta}_c) \boldsymbol{\theta}_\delta \quad (9.13)$$

Forces.

$$\mathbf{f} = \lambda \quad (9.14a)$$

$$\mathbf{m} = \boldsymbol{\Gamma}^{-T}(\boldsymbol{\theta}_c) \bar{\boldsymbol{\mu}} \stackrel{\text{def}}{=} \boldsymbol{\mu} \quad (9.14b)$$

Linearization.

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \Delta x \\ \boldsymbol{\theta}_\Delta \end{Bmatrix} + \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \Delta \lambda \\ \Delta \boldsymbol{\mu} \end{Bmatrix} = - \begin{Bmatrix} -\lambda \\ -\boldsymbol{\mu} \\ -d_c \\ -\boldsymbol{\theta}_c \end{Bmatrix} \quad (9.15)$$

since $\boldsymbol{\Gamma}(\boldsymbol{\theta}_c) \boldsymbol{\theta}_c = \boldsymbol{\theta}_c$.

9.1.2 Distance Joint

The distance joint (`DistanceJoint`) imposes the distance between two nodes, which may depend on time and other states of the problem.

Variants. There is a variant with offsets (`DistanceJointWithOffset`), discussed in Section 9.1.3.

Files. It is implemented in files

`mbdyn/struct/distance.h`
`mbdyn/struct/distance.cc`

Definitions.

$$\mathbf{d} = \mathbf{x}_2 - \mathbf{x}_1 \quad (9.16)$$

$$d = \sqrt{\mathbf{d}^T \mathbf{d}} \quad (9.17)$$

$$\mathbf{u} = \frac{\mathbf{d}}{d} \quad (9.18)$$

Limitations.

$$d > 0 \quad (9.19)$$

Constraint Equation.

$$d \sqrt{\mathbf{u}^T \mathbf{u}} = d \quad (9.20)$$

Forces.

$$\mathbf{F}_1 = \alpha \mathbf{u} \quad (9.21a)$$

$$\mathbf{F}_2 = -\alpha \mathbf{u} \quad (9.21b)$$

Linearization.

$$\begin{bmatrix} \frac{\alpha}{d}\mathbf{I} & -\frac{\alpha}{d}\mathbf{I} & -\mathbf{u} \\ -\frac{\alpha}{d}\mathbf{I} & \frac{\alpha}{d}\mathbf{I} & \mathbf{u} \\ -\mathbf{u}^T & \mathbf{u}^T & 0 \end{bmatrix} \begin{Bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \\ \delta\alpha \end{Bmatrix} = \begin{Bmatrix} \alpha\mathbf{u} \\ -\alpha\mathbf{u} \\ d(1 - \sqrt{\mathbf{u}^T\mathbf{u}}) \end{Bmatrix} \quad (9.22)$$

Constraint Equation Derivative.

$$d\mathbf{u}^T\dot{\mathbf{u}} = 0 \quad (9.23)$$

Force Derivatives.

$$\dot{\mathbf{F}}_1 = \alpha\dot{\mathbf{u}} + \dot{\alpha}\mathbf{u} \quad (9.24a)$$

$$\dot{\mathbf{F}}_2 = -\alpha\dot{\mathbf{u}} - \dot{\alpha}\mathbf{u} \quad (9.24b)$$

where

$$\dot{\mathbf{u}} = \frac{\dot{\mathbf{x}}_2 - \dot{\mathbf{x}}_1}{d} - \mathbf{u}\frac{\dot{d}}{d} \quad (9.25)$$

Linearization.

$$\begin{bmatrix} \frac{\dot{\alpha}}{d}\mathbf{I} & \frac{\alpha}{d}\mathbf{I} & -\frac{\dot{\alpha}}{d}\mathbf{I} & -\frac{\alpha}{d}\mathbf{I} & -\dot{\mathbf{u}} & -\mathbf{u} \\ -\frac{\dot{\alpha}}{d}\mathbf{I} & -\frac{\alpha}{d}\mathbf{I} & \frac{\dot{\alpha}}{d}\mathbf{I} & \frac{\alpha}{d}\mathbf{I} & \dot{\mathbf{u}} & \mathbf{u} \\ -\dot{\mathbf{u}}^T & -\mathbf{u}^T & \dot{\mathbf{u}}^T & \mathbf{u}^T & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \delta\mathbf{x}_1 \\ \delta\dot{\mathbf{x}}_1 \\ \delta\mathbf{x}_2 \\ \delta\dot{\mathbf{x}}_2 \\ \delta\alpha \\ \delta\dot{\alpha} \end{Bmatrix} = \begin{Bmatrix} \alpha\dot{\mathbf{u}} + \dot{\alpha}\mathbf{u} \\ -\alpha\dot{\mathbf{u}} - \dot{\alpha}\mathbf{u} \\ d\mathbf{u}^T\dot{\mathbf{u}} \end{Bmatrix} \quad (9.26)$$

9.1.3 Distance Joint With Offsets

The distance joint with offset (`DistanceJointWithOffset`) imposes the distance between two points that rigidly offset from the respective nodes. It is a variant of the distance joint (`DistanceJoint`) discussed in Section 9.1.2.

Files. It is implemented in files
`mbdyn/struct/distance.h`
`mbdyn/struct/distance.cc`

Definitions.

$$\mathbf{d} = \mathbf{x}_2 + \mathbf{b}_2 - \mathbf{x}_1 - \mathbf{b}_1 \quad (9.27)$$

$$d = \sqrt{\mathbf{d}^T\mathbf{d}} \quad (9.28)$$

$$\mathbf{u} = \frac{\mathbf{d}}{d} \quad (9.29)$$

Limitations:

$$d > 0 \quad (9.30)$$

Constraint equation

$$d\sqrt{\mathbf{u}^T \mathbf{u}} = d \quad (9.31)$$

Forces:

$$\mathbf{F}_1 = \alpha \mathbf{u} \quad (9.32a)$$

$$\mathbf{M}_1 = \alpha \mathbf{b}_1 \times \mathbf{u} \quad (9.32b)$$

$$\mathbf{F}_2 = -\alpha \mathbf{u} \quad (9.32c)$$

$$\mathbf{M}_2 = -\alpha \mathbf{b}_2 \times \mathbf{u} \quad (9.32d)$$

Linearization:

$$\begin{bmatrix} \frac{\alpha}{d} \mathbf{I} & -\frac{\alpha}{d} \mathbf{b}_1 \times & -\frac{\alpha}{d} \mathbf{I} & \frac{\alpha}{d} \mathbf{b}_2 \times & -\mathbf{u} \\ \frac{\alpha}{d} \mathbf{b}_1 \times & -\frac{\alpha}{d} (\mathbf{b}_1 + \mathbf{d}) \times \mathbf{b}_1 \times & -\frac{\alpha}{d} \mathbf{b}_1 \times & \frac{\alpha}{d} \mathbf{b}_1 \times \mathbf{b}_2 \times & -\mathbf{b}_1 \times \mathbf{u} \\ -\frac{\alpha}{d} \mathbf{I} & \frac{\alpha}{d} \mathbf{b}_1 \times & \frac{\alpha}{d} \mathbf{I} & -\frac{\alpha}{d} \mathbf{b}_2 \times & \mathbf{u} \\ -\frac{\alpha}{d} \mathbf{b}_2 \times & \frac{\alpha}{d} \mathbf{b}_2 \times \mathbf{b}_1 \times & \frac{\alpha}{d} \mathbf{b}_2 \times & -\frac{\alpha}{d} (\mathbf{b}_2 - \mathbf{d}) \times \mathbf{b}_2 \times & \mathbf{b}_2 \times \mathbf{u} \\ -\mathbf{u}^T & -(\mathbf{b}_1 \times \mathbf{u})^T & \mathbf{u}^T & (\mathbf{b}_2 \times \mathbf{u})^T & 0 \end{bmatrix} \begin{Bmatrix} \delta \mathbf{x}_1 \\ \delta \mathbf{g}_1 \\ \delta \mathbf{x}_2 \\ \delta \mathbf{g}_2 \\ \delta \alpha \end{Bmatrix} = \begin{Bmatrix} \alpha \mathbf{u} \\ \alpha \mathbf{b}_1 \times \mathbf{u} \\ -\alpha \mathbf{u} \\ -\alpha \mathbf{b}_2 \times \mathbf{u} \\ d(1 - \sqrt{\mathbf{u}^T \mathbf{u}}) \end{Bmatrix} \quad (9.33)$$

Constraint Equation Derivative

$$d\mathbf{u}^T \dot{\mathbf{u}} = 0 \quad (9.34)$$

Forces:

$$\dot{\mathbf{F}}_1 = \alpha \dot{\mathbf{u}} + \dot{\alpha} \mathbf{u} \quad (9.35a)$$

$$\dot{\mathbf{M}}_1 = \alpha (\boldsymbol{\omega}_1 \times \mathbf{b}_1) \times \mathbf{u} + \alpha \mathbf{b}_1 \times \dot{\mathbf{u}} + \dot{\alpha} \mathbf{b}_1 \times \mathbf{u} \quad (9.35b)$$

$$\dot{\mathbf{F}}_2 = -\alpha \dot{\mathbf{u}} - \dot{\alpha} \mathbf{u} \quad (9.35c)$$

$$\dot{\mathbf{M}}_2 = -\alpha (\boldsymbol{\omega}_2 \times \mathbf{b}_2) \times \mathbf{u} - \alpha \mathbf{b}_2 \times \dot{\mathbf{u}} - \dot{\alpha} \mathbf{b}_2 \times \mathbf{u} \quad (9.35d)$$

Linearization: TODO(C).

9.1.4 Spherical hinge

The spherical hinge joint (`SphericalHingeJoint`) constrains the positions of two points, that may be rigidly offset from two nodes, to be coincident.

Variants. There exists a pinned version (`SphericalPinJoint`, TODO©).

Files. It is implemented in files

`mbdyn/struct/spherj.h`
`mbdyn/struct/spherj.cc`

Joint data.

$$\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2 \quad (9.36)$$

where:

$\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2$ offset of connection point from nodes 1, 2 in node reference;
 Constraint equations (normalized¹ by `dCoef`)

$$(\mathbf{x}_2 + \mathbf{b}_2) - (\mathbf{x}_1 + \mathbf{b}_1) = \mathbf{0} \quad (9.37)$$

where:

$\mathbf{x}_1, \mathbf{x}_2$ position of nodes 1, 2;
 $\mathbf{R}_1, \mathbf{R}_2$ orientation of nodes 1, 2.

$\mathbf{b}_1 = \mathbf{R}_1 \tilde{\mathbf{b}}_1, \mathbf{b}_2 = \mathbf{R}_2 \tilde{\mathbf{b}}_2$ offset of connection point from nodes 1, 2 in global frame
 Residual vector:

$$\begin{aligned} \text{node1 momentum : } 1 - 3 &= \mathbf{F} \\ \text{node1 angular momentum : } 4 - 6 &= (\mathbf{R}_1 \cdot \mathbf{d}_1) \times \mathbf{F} \\ \text{node2 momentum : } 7 - 9 &= \mathbf{F} \\ \text{node2 angular momentum : } 10 - 12 &= (\mathbf{R}_2 \cdot \mathbf{d}_2) \times \mathbf{F} \\ \text{constraint : } 13 - 15 &= ((\mathbf{x}_1 + \mathbf{R}_1 \cdot \mathbf{d}_1) - (\mathbf{x}_2 + \mathbf{R}_2 \cdot \mathbf{d}_2)) / \text{dCoef} \end{aligned}$$

where:

\mathbf{F} constraint reaction force.

9.1.5 Revolute hinge

The revolute hinge joint (`PlaneHingeJoint`) constrains the positions of two points, that may be rigidly offset from two nodes, to be coincident. It also constrains their orientations to keep the respective axis 3 parallel.

Variants. There exist a pinned version (`RevolutePinJoint`, TODO©), a version that does not constrain the position (`RevoluteRotationJoint`), and a version which imposes the relative angular velocity about axis 3 (`AxialRotationJoint`).

Files. It is implemented in files

`mbdyn/struct/planej.h`
`mbdyn/struct/planej.cc`

¹When purely algebraic constraints are considered, to improve the scaling of the matrix, the constraint equation can be divided by `dCoef`, the coefficient related to the integration method illustrated in Equation (6.27).

Joint data.

$$\mathbf{d}_1, \mathbf{d}_2, \mathbf{R}_{h1}, \mathbf{R}_{h2} \quad (9.38)$$

where:

$\mathbf{d}_1, \mathbf{d}_2$: offset of nodes 1,2 in node reference;
 $\mathbf{R}_{h1}, \mathbf{R}_{h2}$: joint relative orientation wrt. nodes 1,2 (FIXME).

Constraint equations (normalized by **dCofef**)

$$\begin{aligned} (\mathbf{x}_1 + \mathbf{R}_1 \cdot \mathbf{d}_1) - (\mathbf{x}_2 + \mathbf{R}_1 \cdot \mathbf{d}_2) &= 0 \\ (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \cdot (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[2] &= 0 \\ (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \cdot (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[1] &= 0 \end{aligned}$$

where:

$\mathbf{x}_1, \mathbf{x}_2$: positions of nodes 1,2;
 $\mathbf{R}_1, \mathbf{R}_2$: orientation of nodes 1, 2.

Residual vector:

$$\begin{aligned} \text{node1 momentum : } 1 - 3 &- = \mathbf{F} \\ \text{node1 angular momentum : } 4 - 6 &- = (\mathbf{R}_1 \cdot \mathbf{d}_1) \times \mathbf{F} + \\ &\quad (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[2] \times (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] * \mathbf{M}[1] + \\ &\quad (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \times (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[1] * \mathbf{M}[2] \\ \text{node2 momentum : } 7 - 9 &+ = \mathbf{F} \\ \text{node2 angular momentum : } 10 - 12 &+ = (\mathbf{R}_2 \cdot \mathbf{d}_2) \times \mathbf{F} + \\ &\quad (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[2] \times (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] * \mathbf{M}[1] + \\ &\quad (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \times (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[1] * \mathbf{M}[2] \\ \text{constraint : } 13 - 15 &= ((\mathbf{x}_1 + \mathbf{R}_1 \cdot \mathbf{d}_1) - (\mathbf{x}_2 + \mathbf{R}_1 \cdot \mathbf{d}_2)) / \text{dCofef} \\ \text{constraint : } 16 &= ((\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \cdot (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[2]) / \text{dCofef} \\ \text{constraint : } 17 &= ((\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \cdot (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[1]) / \text{dCofef} \end{aligned}$$

where:

\mathbf{F} : constraint reaction force;
 \mathbf{M} : constraint moment reaction (third component null).

Friction:

- add third component of constraint moment \mathbf{M}
- add a constraint equation; this could be one of the following:
 - direct definition of $\mathbf{M}[3]$ in function of relative velocity, friction coefficient and \mathbf{F}
 - impose null relative velocity
- optionally add internal states dynamic z (for friction)

add third component of constraint moment M

Residual vector:

$$\begin{aligned}
\text{node1 momentum : } 1 - 3 & - = \mathbf{F} \\
\text{node1 angular momentum : } 4 - 6 & - = (\mathbf{R}_1 \cdot \mathbf{d}_1) \times \mathbf{F} + \\
& (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[2] \times (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] * M[1] + \\
& (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \times (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[1] * M[2] + \\
& (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] * M[3] \\
\text{node2 momentum : } 7 - 9 & + = \mathbf{F} \\
\text{node2 angular momentum : } 10 - 12 & + = (\mathbf{R}_2 \cdot \mathbf{d}_2) \times \mathbf{F} + \\
& (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[2] \times (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] * M[1] + \\
& (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \times (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[1] * M[2] + \\
& (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] * M[3] \\
\text{constraint : } 13 - 15 & = ((\mathbf{x}_1 + \mathbf{R}_1 \cdot \mathbf{d}_1) - (\mathbf{x}_2 + \mathbf{R}_1 \cdot \mathbf{d}_2)) / dCoef \\
\text{constraint : } 16 & = ((\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \cdot (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[2]) / dCoef \\
\text{constraint : } 17 & = ((\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \cdot (\mathbf{R}_2 \cdot \mathbf{R}_{h2})[1]) / dCoef \\
\text{friction : } 18 & = M[3] - f(\mathbf{F}, v, \text{friction coef}) \\
& = (\text{rel velocity})[3] \\
(\text{optional}) \text{ friction states : } 19... & = \dot{z} - g(z, v)
\end{aligned}$$

where v is the relative velocity $v = r * (\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2) \cdot (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3]$. The constraint is based on positions. This means that during integration $(\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2)$ will NOT have exactly the direction $(\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3]$. We choose to disregard this error. The friction moment should be along $(\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3]$.

r is the joint radius.

CHECK THIS!!!

Explicit form of friction moment contribution (FIXME: remove vector $(\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3]$ and write scalar equation???:

$f(\mathbf{F}, \text{rel velocity}) : M[3] = sh_c(||\mathbf{F}||, f_c(v, z, \dot{z})) * (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] * ||\mathbf{F}|| * f_c(v, z, \dot{z})$,
with \mathbf{R}_{h1} constant.

Variation of friction moment contribution:

$$\begin{aligned}
\delta(M[3](\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3]) & = M[3] * (\mathbf{R}_{\delta 1} \times \mathbf{R}_1)[3] + \\
& (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] * \delta M[3] \\
& = M[3] * \mathbf{R}_1^T \cdot (\mathbf{R}_{\delta 1} \times)[3] + \\
& (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] * \delta M[3]
\end{aligned}$$

Variation of friction moment contribution component:

$$\begin{aligned}
\delta \mathbf{M}[3] &= \text{sh_c}(\|\mathbf{F}\|, f_c(v, z, \dot{z})) * f_c(v, z, \dot{z}) * \frac{\mathbf{F}}{\|\mathbf{F}\|} \cdot \delta \mathbf{F} + \\
&\quad \text{sh_c}(\|\mathbf{F}\|, f_c(v, z, \dot{z})) * (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] * \|\mathbf{F}\| * \frac{\partial f_c}{\partial v} * \delta v + \\
&\quad \|\mathbf{F}\| * f_c(v, z, \dot{z}) * \frac{\partial \text{sh_c}(\|\mathbf{F}\|, f_c(v, z, \dot{z}))}{\partial \|\mathbf{F}\|} \frac{\partial \|\mathbf{F}\|}{\partial \mathbf{F}} \cdot \delta \mathbf{F} + \\
&\quad \|\mathbf{F}\| * f_c(v, z, \dot{z}) * \frac{\partial \text{sh_c}(\|\mathbf{F}\|, f_c(v, z, \dot{z}))}{\partial f_c(v, z, \dot{z})} * \frac{\partial f_c}{\partial v} * \delta v \\
&= \left(\begin{array}{l} \text{sh_c}(\|\mathbf{F}\|, f_c(v, z, \dot{z})) * f_c(v, z, \dot{z}) \frac{\mathbf{F}}{\|\mathbf{F}\|} \\ \|\mathbf{F}\| * f_c(v, z, \dot{z}) * \frac{\partial \text{sh_c}(\|\mathbf{F}\|, f_c(v, z, \dot{z}))}{\partial \|\mathbf{F}\|} \end{array} \right) \cdot \delta \mathbf{F} + \\
&\quad \left(\begin{array}{l} \text{sh_c}(\|\mathbf{F}\|, f_c(v, z, \dot{z})) * \|\mathbf{F}\| \\ \|\mathbf{F}\| * f_c(v, z, \dot{z}) * \frac{\partial \text{sh_c}(\|\mathbf{F}\|, f_c(v, z, \dot{z}))}{\partial f_c(v, z, \dot{z})} \end{array} \right) * \left\{ \begin{array}{l} \frac{\partial f_c}{\partial v} * \delta v \\ \frac{\partial f_c}{\partial z} * \delta z \\ \frac{\partial f_c}{\partial \dot{z}} * \delta \dot{z} \end{array} \right\}
\end{aligned}$$

where

- $v = r * (\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2) \cdot (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3]$ so that

$$\begin{aligned}
\delta v &= d * (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \cdot (\delta \boldsymbol{\omega}_1 - \delta \boldsymbol{\omega}_2) + \\
&\quad d * (\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2) \cdot (\mathbf{R}_{\delta 1} \times \mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \\
&= d * (\mathbf{R}_1 \cdot \mathbf{R}_{h1})[3] \cdot (\delta \boldsymbol{\omega}_1 - \delta \boldsymbol{\omega}_2) + \\
&\quad d * (\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2) \cdot \mathbf{R}_{h1}^T \cdot \mathbf{R}_1^T \cdot (\mathbf{R}_{\delta 1} \times)[3]
\end{aligned}$$

- $\text{sh_c}(\|\mathbf{F}\|, f_c(v, z, \dot{z})) = r * C_\alpha(\alpha(\text{constants}, f_c(v, z, \dot{z}), \|\mathbf{F}\|, f_c(v, z, \dot{z}))) \frac{1}{\sqrt{1 + f_c^2(v, z, \dot{z})}}$

- $\alpha(\text{constants}, f_c(v, z, \dot{z}), \|\mathbf{F}\|, f_c(v, z, \dot{z})) = \sin^{-1} \left(\sqrt{\frac{2 * 31 * \|\mathbf{F}\|}{E * b * \sqrt{1 + f_c^2(v, z, \dot{z})}} \frac{r'/r}{r' - r}} \right)$

- for very low joint loads (angle of contact $\alpha < 20^\circ$, i.e. about less than 1% of the joint allowable load) we can safely assume $C_\alpha \approx 1$ so that

$$\text{sh_c}(\|\mathbf{F}\|, f_c(v, z, \dot{z})) \approx r * \frac{1}{\sqrt{1 + f_c^2(v, z, \dot{z})}},$$

$$\frac{\partial \text{sh_c}(\|\mathbf{F}\|, f_c(v, z, \dot{z}))}{\partial \|\mathbf{F}\|} = \mathbf{0}$$

and

$$\frac{\partial \text{sh_c}(\|\mathbf{F}\|, f_c(v, z, \dot{z}))}{\partial f_c(v, z, \dot{z})} = -r * (1 + f_c^2(v, z, \dot{z}))^{-3/2} * f_c(v, z, \dot{z})$$

9.1.6 Inline

This joint forces a point rigidly attached to node b to slide along a line rigidly attached to node a , thus removing 2 degrees of freedom.

Files. It is implemented in files

`mbdyn/struct/inline.h`
`mbdyn/struct/inline.cc`

Definitions. Orientation of line

$$\mathbf{R}_v = \mathbf{R}_a \tilde{\mathbf{R}}_v \quad (9.39)$$

the axis is along local axis 3 of matrix $\tilde{\mathbf{R}}_v$, $\tilde{\mathbf{e}}_3$, while directions $\tilde{\mathbf{e}}_1$, $\tilde{\mathbf{e}}_2$ are constrained.

Origin of line

$$\mathbf{x}_p = \mathbf{x}_a + \mathbf{p} = \mathbf{x}_a + \mathbf{R}_a \tilde{\mathbf{p}} \quad (9.40)$$

Sliding point

$$\mathbf{x}_q = \mathbf{x}_b + \mathbf{q} = \mathbf{x}_b + \mathbf{R}_b \tilde{\mathbf{q}} \quad (9.41)$$

Constraint equations: distance between sliding point \mathbf{x}_q and axis null in local directions $\tilde{\mathbf{e}}_1$, $\tilde{\mathbf{e}}_2$, namely

$$\mathbf{e}_i^T \mathbf{d} = 0, \quad (9.42)$$

with $i = 1, 2$ and $\mathbf{d} = \mathbf{x}_q - \mathbf{x}_p$. After simple manipulation,

$$\mathbf{e}_i^T (\mathbf{x}_b + \mathbf{q} - \mathbf{x}_a) - \tilde{\mathbf{e}}_i^T \tilde{\mathbf{p}} = 0. \quad (9.43)$$

Perturbation:

$$(\mathbf{e}_i \times (\mathbf{x}_b + \mathbf{q} - \mathbf{x}_a))^T \boldsymbol{\theta}_{a\delta} + \mathbf{e}_i^T (\delta \mathbf{x}_b - \mathbf{q} \times \boldsymbol{\theta}_{b\delta} - \delta \mathbf{x}_a) = 0 \quad (9.44)$$

Forces and moments:

$$\mathbf{f}_a = - \sum_{i=1,2} \mathbf{e}_i \lambda_i \quad (9.45a)$$

$$\mathbf{m}_a = \sum_{i=1,2} \mathbf{e}_i \times (\mathbf{x}_b + \mathbf{q} - \mathbf{x}_a) \lambda_i \quad (9.45b)$$

$$\mathbf{f}_b = \sum_{i=1,2} \mathbf{e}_i \lambda_i \quad (9.45c)$$

$$\mathbf{m}_b = - \sum_{i=1,2} \mathbf{e}_i \times \mathbf{q} \lambda_i \quad (9.45d)$$

Linearization

$$\left[\begin{array}{cc} -\mathbf{e}_i^T & (\mathbf{e}_i \times (\mathbf{x}_b + \mathbf{q} - \mathbf{x}_a))^T \end{array} \right] \left\{ \begin{array}{c} \delta \mathbf{x}_a \\ \boldsymbol{\theta}_{a\delta} \\ \delta \mathbf{x}_b \\ \boldsymbol{\theta}_{b\delta} \end{array} \right\} = 0 \quad (9.46)$$

$$\delta \mathbf{f}_a = - \sum_{i=1,2} (\mathbf{e}_i \delta \lambda_i - \lambda_i \mathbf{e}_i \times \boldsymbol{\theta}_{a\delta}) \quad (9.47a)$$

$$\delta \mathbf{m}_a = \sum_{i=1,2} (\mathbf{e}_i \times (\mathbf{x}_b + \mathbf{q} - \mathbf{x}_a) \delta \lambda_i + \lambda_i (\mathbf{x}_b + \mathbf{q} - \mathbf{x}_a) \times \mathbf{e}_i \times \boldsymbol{\theta}_{a\delta} + \lambda_i \mathbf{e}_i \times (\delta \mathbf{x}_b - \mathbf{q} \times \boldsymbol{\theta}_{b\delta} - \delta \mathbf{x}_a)) \quad (9.47b)$$

$$\delta \mathbf{f}_b = \sum_{i=1,2} (\mathbf{e}_i \delta \lambda_i - \lambda_i \mathbf{e}_i \times \boldsymbol{\theta}_{a\delta}) \quad (9.47c)$$

$$\delta \mathbf{m}_b = - \sum_{i=1,2} (\mathbf{e}_i \times \mathbf{q} \delta \lambda_i + \lambda_i \mathbf{q} \times \mathbf{e}_i \times \boldsymbol{\theta}_{a\delta} - \lambda_i \mathbf{e}_i \times \mathbf{q} \times \boldsymbol{\theta}_{b\delta}) \quad (9.47d)$$

or

$$\begin{aligned} \left\{ \begin{array}{l} \delta \mathbf{f}_a \\ \delta \mathbf{m}_a \\ \delta \mathbf{f}_b \\ \delta \mathbf{m}_b \end{array} \right\} &= \begin{bmatrix} \mathbf{e}_1 & -\mathbf{e}_2 \\ \mathbf{e}_1 \times (\mathbf{x}_b + \mathbf{q} - \mathbf{x}_a) & \mathbf{e}_2 \times (\mathbf{x}_b + \mathbf{q} - \mathbf{x}_a) \\ \mathbf{e}_1 & \mathbf{e}_2 \\ -\mathbf{e}_1 \times \mathbf{q} & -\mathbf{e}_2 \times \mathbf{q} \end{bmatrix} \left\{ \begin{array}{l} \delta \lambda_1 \\ \delta \lambda_2 \end{array} \right\} \\ &+ \begin{bmatrix} \mathbf{0} & \sum_{i=1,2} \lambda_i \mathbf{e}_i \times & \mathbf{0} & \mathbf{0} \\ -\sum_{i=1,2} \lambda_i \mathbf{e}_i \times & \sum_{i=1,2} \lambda_i (\mathbf{x}_b + \mathbf{q} - \mathbf{x}_a) \times \mathbf{e}_i \times & \sum_{i=1,2} \lambda_i \mathbf{e}_i \times & -\sum_{i=1,2} \lambda_i \mathbf{e}_i \times \mathbf{q} \times \\ \mathbf{0} & -\sum_{i=1,2} \lambda_i \mathbf{e}_i \times & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\sum_{i=1,2} \lambda_i \mathbf{q} \times \mathbf{e}_i \times & \mathbf{0} & \sum_{i=1,2} \lambda_i \mathbf{e}_i \times \mathbf{q} \times \end{bmatrix} \left\{ \begin{array}{l} \delta \mathbf{x}_a \\ \boldsymbol{\theta}_{a\delta} \\ \delta \mathbf{x}_b \\ \boldsymbol{\theta}_{b\delta} \end{array} \right\} \end{aligned} \quad (9.48)$$

9.1.7 Drive Hinge

This joint (`DriveHingeJoint`) forces two nodes to assume a relative orientation given by a rotation vector $\boldsymbol{\theta}$, whose direction with respect to node 1 represents the rotation axis, and whose amplitude represents the magnitude of the rotation.

Variants. There exist no variants, although a pinned version may be of use (TODO©).

Files. It is implemented in files

`mbdyn/struct/drvhinge.h`
`mbdyn/struct/drvhinge.cc`

Definitions.

$$\begin{aligned} \mathbf{R}_{\text{rel}} &= \mathbf{R}_1^T \mathbf{R}_2 \\ \boldsymbol{\theta} &= \text{ax}(\exp^{-1}(\mathbf{R}_{\text{rel}})) \end{aligned}$$

Limitations:

$$\|\boldsymbol{\theta}\| < \pi \quad (9.49)$$

Constraint equation

$$\boldsymbol{\theta} - \boldsymbol{\theta}_0 = \mathbf{0} \quad (9.50)$$

Couples:

$$\begin{aligned}\mathbf{M}_1 &= -\mathbf{R}_1 \boldsymbol{\alpha} \\ \mathbf{M}_2 &= \mathbf{R}_1 \boldsymbol{\alpha}\end{aligned}$$

Linearization:

$$\begin{bmatrix} (\mathbf{R}_1 \boldsymbol{\alpha}) \times & 0 & -\mathbf{R}_1 \\ -(\mathbf{R}_1 \boldsymbol{\alpha}) \times & 0 & \mathbf{R}_1 \\ -\boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_1^T & \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_1^T & 0 \end{bmatrix} \begin{Bmatrix} \delta \mathbf{g}_1 \\ \delta \mathbf{g}_2 \\ \delta \boldsymbol{\alpha} \end{Bmatrix} = \begin{Bmatrix} \mathbf{R}_1 \boldsymbol{\alpha} \\ -\mathbf{R}_1 \boldsymbol{\alpha} \\ \boldsymbol{\theta}_0 - \boldsymbol{\theta} \end{Bmatrix} \quad (9.51)$$

The linearization of the reaction moments contribution to the moment equilibrium equations of the nodes is straightforward. The linearization of the constraint equation is a bit more complicated. According to the definition of $\boldsymbol{\theta}$, its linearization yields

$$\begin{aligned}\delta \boldsymbol{\theta} &= \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \text{ax}(\delta \mathbf{R}_{\text{rel}} \mathbf{R}_{\text{rel}}^T) \\ &= \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \text{ax}(\delta \mathbf{R}_1^T \mathbf{R}_1 + \mathbf{R}_1^T \delta \mathbf{R}_2 \mathbf{R}_2^T \mathbf{R}_1) \\ &= \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \text{ax}(\mathbf{R}_1^T \boldsymbol{\theta}_{1\delta} \times^T \mathbf{R}_1 + \mathbf{R}_1^T \boldsymbol{\theta}_{2\delta} \times \mathbf{R}_1) \\ &= \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \text{ax}(-\mathbf{R}_1^T \boldsymbol{\theta}_{1\delta} \times \mathbf{R}_1 + \mathbf{R}_1^T \boldsymbol{\theta}_{2\delta} \times \mathbf{R}_1) \\ &= \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \text{ax}(\mathbf{R}_1^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) \times \mathbf{R}_1) \\ &= \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_1^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta})\end{aligned}$$

according to the updated-updated² simplifications, $\boldsymbol{\theta}_{i\delta} \stackrel{\text{uu}}{=} \delta \mathbf{g}_i$, thus resulting in

$$\boldsymbol{\theta}_\delta = \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_1^T (\delta \mathbf{g}_2 - \delta \mathbf{g}_1) \quad (9.52)$$

note that $\boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1}$ does not simplify to \mathbf{I} because $\boldsymbol{\theta}$ in general is a finite rotation.

9.1.8 Drive Displacement

This joint (`DriveDisplacementJoint`) constrains the relative position of a point optionally offset from node b with respect to a point optionally offset from node a , so that their relative position matches a value that is imposed in node a reference frame; the imposed vector may depend on time and other states during the simulation.

Variants. There exists a pinned version (`DriveDisplacementPinJoint`), discussed in Section 9.1.9.

Files. It is implemented in files

```
mbdyn/struct/drvdisp.h
mbdyn/struct/drvdisp.cc
```

Constraint Equation. The constraint equation is

$$\boldsymbol{\phi} = \mathbf{x}_b + \mathbf{f}_b - \mathbf{x}_a - \mathbf{f}_a - \mathbf{v} = \mathbf{0} \quad (9.53)$$

or, based on the initial assumptions,

$$\boldsymbol{\phi} = \mathbf{x}_b + \mathbf{f}_b - \mathbf{x}_a - \mathbf{R}_a (\tilde{\mathbf{f}}_a + \tilde{\mathbf{v}}) = \mathbf{0} \quad (9.54)$$

²In the following, the operator $\stackrel{\text{uu}}{=}$ will be used to indicate the updated-updated approximation.

The perturbation of the constraint equation yields

$$\delta\phi = \delta\mathbf{x}_b - \mathbf{f}_b \times \boldsymbol{\theta}_{b\delta} - \delta\mathbf{x}_a + \mathbf{d} \times \boldsymbol{\theta}_{a\delta} \quad (9.55)$$

where $\mathbf{d} = \mathbf{R}_a (\tilde{\mathbf{f}}_a + \tilde{\mathbf{v}})$. The contribution of the constraint to the equilibrium equations is

$$\begin{Bmatrix} F_a \\ C_a \\ F_b \\ C_b \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{\lambda} \\ -\mathbf{d} \times \boldsymbol{\lambda} \\ \boldsymbol{\lambda} \\ \mathbf{f}_b \times \boldsymbol{\lambda} \end{Bmatrix} \quad (9.56)$$

where $\boldsymbol{\lambda}$ is the reaction force in the global reference frame.

The contribution of forces and moments and of the constraint equation to the Jacobian matrix is

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & -\boldsymbol{\lambda} \times \mathbf{d} \times & \mathbf{0} & \mathbf{0} & -\mathbf{d} \times \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\lambda} \times \mathbf{f}_b \times & \mathbf{f}_b \times \\ -\mathbf{I} & \mathbf{d} \times & \mathbf{I} & -\mathbf{f}_b \times & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \delta\mathbf{x}_a \\ \boldsymbol{\theta}_{a\delta} \\ \delta\mathbf{x}_b \\ \boldsymbol{\theta}_{b\delta} \\ \delta\boldsymbol{\lambda} \end{Bmatrix} \quad (9.57)$$

The derivative of the constraint equation is

$$\dot{\phi} = \dot{\mathbf{x}}_b + \boldsymbol{\omega}_b \times \mathbf{f}_b - \dot{\mathbf{x}}_a - \boldsymbol{\omega}_a \times \mathbf{d} - \dot{\mathbf{d}} \quad (9.58)$$

where $\dot{\mathbf{d}} = \mathbf{R}_a \dot{\tilde{\mathbf{v}}}$; its perturbation yields

$$\delta\dot{\phi} = \delta\dot{\mathbf{x}}_b + \delta\boldsymbol{\omega}_b \times \mathbf{f}_b + \boldsymbol{\omega}_b \times \boldsymbol{\theta}_{b\delta} \times \mathbf{f}_b - \delta\dot{\mathbf{x}}_a - \delta\boldsymbol{\omega}_a \times \mathbf{d} - \boldsymbol{\omega}_a \times \boldsymbol{\theta}_{a\delta} \times \mathbf{d} - \boldsymbol{\theta}_{a\delta} \times \dot{\mathbf{d}}. \quad (9.59)$$

The contributions to the Jacobian matrix and residual of the initial assembly are

$$\begin{aligned}
& \left[\begin{array}{ccccc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & -\boldsymbol{\lambda} \times \mathbf{d} \times -\boldsymbol{\mu} \times \dot{\mathbf{d}} \times +(\boldsymbol{\omega}_a \times \boldsymbol{\mu}) \times \mathbf{d} \times & \mathbf{0} & \mathbf{0} & -\mathbf{d} \times \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\lambda} \times \mathbf{f}_b \times -(\boldsymbol{\omega}_b \times \boldsymbol{\mu}) \times \mathbf{f}_b \times & \mathbf{f}_b \times \\ -\mathbf{I} & \mathbf{d} \times & \mathbf{I} & -\mathbf{f}_b \times & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{\mu} \times \mathbf{d} \times & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mu} \times \mathbf{f}_b \times & \mathbf{0} \\ \mathbf{0} & \dot{\mathbf{d}} \times +\boldsymbol{\omega}_a \times \mathbf{d} \times & \mathbf{0} & -\boldsymbol{\omega}_b \times \mathbf{f}_b \times & \mathbf{0} \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_a \\ \theta_{a\delta} \\ \delta \mathbf{x}_b \\ \theta_{b\delta} \\ \delta \boldsymbol{\lambda} \end{array} \right\} \\
& + \left[\begin{array}{ccccc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{d} \times \boldsymbol{\mu} \times & \mathbf{0} & \mathbf{0} & -\dot{\mathbf{d}} \times +\mathbf{d} \times \boldsymbol{\omega}_a \times \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{f}_b \times \boldsymbol{\mu} \times & -\mathbf{f}_b \times \boldsymbol{\omega}_b \times \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{d} \times \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{f}_b \times \\ -\mathbf{I} & \mathbf{d} \times & \mathbf{I} & -\mathbf{f}_b \times & \mathbf{0} \end{array} \right] \left\{ \begin{array}{l} \delta \dot{\mathbf{x}}_a \\ \delta \boldsymbol{\omega}_a \\ \delta \dot{\mathbf{x}}_b \\ \delta \boldsymbol{\omega}_b \\ \delta \boldsymbol{\mu} \end{array} \right\} \\
& = \left\{ \begin{array}{l} \boldsymbol{\lambda} \\ \mathbf{d} \times \boldsymbol{\lambda} - \mathbf{d} \times \boldsymbol{\omega}_a \times \boldsymbol{\mu} + \dot{\mathbf{d}} \times \boldsymbol{\mu} \\ -\boldsymbol{\lambda} \\ -\mathbf{f}_b \times \boldsymbol{\lambda} + \mathbf{f}_b \times \boldsymbol{\omega}_b \times \boldsymbol{\mu} \\ \mathbf{x}_a + \mathbf{d} - \mathbf{x}_b - \mathbf{f}_b \\ \boldsymbol{\mu} \\ \boldsymbol{\mu} \\ -\mathbf{f}_b \times \boldsymbol{\mu} \\ \dot{\mathbf{x}}_a + \boldsymbol{\omega}_a \times \mathbf{d} + \dot{\mathbf{d}} - \dot{\mathbf{x}}_b - \boldsymbol{\omega}_b \times \mathbf{f}_b \end{array} \right\} \quad (9.60)
\end{aligned}$$

9.1.9 Drive Displacement Pin

This joint (`DriveDisplacementPinJoint`) is the “pinned” variant of the drive displacement joint (`DriveDisplacementJoint`) discussed in Section 9.1.8. It imposes the absolute position \mathbf{v} of a point optionally connected to a node by a rigid offset \mathbf{f} :

$$\boldsymbol{\phi} = \mathbf{x} + \mathbf{f} - \mathbf{x}_0 - \mathbf{v} = \mathbf{0} \quad (9.61)$$

The perturbation of the constraint equation yields

$$\delta \boldsymbol{\phi} = \delta \mathbf{x} - \mathbf{f} \times \boldsymbol{\theta}_\delta \quad (9.62)$$

The contribution of the constraint to the equilibrium equations is

$$\left\{ \begin{array}{l} \mathbf{F} \\ \mathbf{C} \end{array} \right\} = \left\{ \begin{array}{l} \boldsymbol{\lambda} \\ \mathbf{f} \times \boldsymbol{\lambda} \end{array} \right\} \quad (9.63)$$

where $\boldsymbol{\lambda}$ is the reaction force in the absolute reference frame.

The contribution of forces and moments and of the constraint equation to the Jacobian matrix is

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \boldsymbol{\lambda} \times \mathbf{f} \times & \mathbf{f} \times \\ \mathbf{I} & -\mathbf{f} \times & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \delta \mathbf{x} \\ \boldsymbol{\theta}_\delta \\ \delta \boldsymbol{\lambda} \end{Bmatrix} \quad (9.64)$$

The derivative of the constraint equation is

$$\dot{\phi} = \dot{\mathbf{x}} + \boldsymbol{\omega} \times \mathbf{f} - \dot{\mathbf{v}}; \quad (9.65)$$

its perturbation yields

$$\delta \dot{\phi} = \delta \dot{\mathbf{x}} - \mathbf{f} \times \delta \boldsymbol{\omega} - \boldsymbol{\omega} \times \mathbf{f} \times \boldsymbol{\theta}_\delta \quad (9.66)$$

The contributions to the Jacobian matrix and residual of the initial assembly are

$$\begin{aligned} & \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \boldsymbol{\lambda} \times \mathbf{f} \times - (\boldsymbol{\omega} \times \boldsymbol{\mu}) \times \mathbf{f} \times & \mathbf{f} \times \\ \mathbf{I} & -\mathbf{f} \times & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \delta \mathbf{x} \\ \boldsymbol{\theta}_\delta \\ \delta \boldsymbol{\lambda} \end{Bmatrix} \\ & \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f} \times \boldsymbol{\mu} \times & -\mathbf{f} \times \boldsymbol{\omega} \times \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \delta \dot{\mathbf{x}} \\ \delta \boldsymbol{\omega} \\ \delta \boldsymbol{\mu} \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{\lambda} \\ -\mathbf{f} \times \boldsymbol{\lambda} + \mathbf{f} \times \boldsymbol{\omega} \times \boldsymbol{\mu} \\ \mathbf{x}_0 + \mathbf{v} - \mathbf{x} - \mathbf{f} \\ -\boldsymbol{\mu} \\ -\mathbf{f} \times \boldsymbol{\mu} \\ \dot{\mathbf{v}} - \dot{\mathbf{x}} - \boldsymbol{\omega} \times \mathbf{f} \end{Bmatrix} \end{aligned} \quad (9.67)$$

9.1.10 Imposed Displacement

The imposed displacement joint (`ImposedDisplacementJoint`) is similar to the drive displacement joint (`DriveDisplacementPinJoint`), discussed in Section 9.1.9, but the relative displacement v , a scalar possibly dependent on the time and on other states of the problem, is imposed only in direction $\tilde{\mathbf{e}}_a$, relative to node a .

Variants. There exists a pinned version (`ImposedDisplacementPinJoint`), discussed in Section 9.1.11.

Files. It is implemented in files
`mbdyn/struct/impdisp.h`
`mbdyn/struct/impdisp.cc`

Constraint equation.

$$\phi = \mathbf{e}_a^T (\mathbf{x}_b + \mathbf{f}_b - \mathbf{x}_a - \mathbf{f}_a) - v = 0, \quad (9.68)$$

or, based on the initial assumptions,

$$\phi = \mathbf{e}_a^T (\mathbf{x}_b + \mathbf{f}_b - \mathbf{x}_a) - (\tilde{\mathbf{e}}_a^T \tilde{\mathbf{f}}_a + v) = 0, \quad (9.69)$$

or

$$\phi = \mathbf{e}_a^T \mathbf{d} - (\tilde{\mathbf{e}}_a^T \tilde{\mathbf{f}}_a + v) = 0, \quad (9.70)$$

where $\mathbf{d} = \mathbf{x}_b + \mathbf{f}_b - \mathbf{x}_a$ and $\tilde{\mathbf{e}}_a$ is constant in the reference frame of node a . The perturbation of the constraint equation yields

$$\delta\phi = \mathbf{e}_a^T (\delta\mathbf{x}_b - \mathbf{f}_b \times \boldsymbol{\theta}_{b\delta} - \delta\mathbf{x}_a) + (\mathbf{e}_a \times (\mathbf{x}_b + \mathbf{f}_b - \mathbf{x}_a))^T \boldsymbol{\theta}_{a\delta}, \quad (9.71)$$

or

$$\delta\phi = \mathbf{e}_a^T (\delta\mathbf{x}_b - \mathbf{f}_b \times \boldsymbol{\theta}_{b\delta} - \delta\mathbf{x}_a) + (\mathbf{e}_a \times \mathbf{d})^T \boldsymbol{\theta}_{a\delta}. \quad (9.72)$$

The contribution of the constraint to the equilibrium equations is

$$\left\{ \begin{array}{c} \mathbf{F}_a \\ \mathbf{C}_a \\ \mathbf{F}_b \\ \mathbf{C}_b \end{array} \right\} = \left\{ \begin{array}{c} -\mathbf{e}_a \lambda \\ -\mathbf{d} \times \mathbf{e}_a \lambda \\ \mathbf{e}_a \lambda \\ \mathbf{f}_b \times \mathbf{e}_a \lambda \end{array} \right\} \quad (9.73)$$

where λ is the reaction force (a scalar).

The contribution of forces and moments and of the constraint equation to the Jacobian matrix is

$$\left[\begin{array}{ccccc} \mathbf{0} & \lambda \mathbf{e}_a \times & \mathbf{0} & \mathbf{0} & -\mathbf{e}_a \\ -\lambda \mathbf{e}_a \times & \lambda \mathbf{d} \times \mathbf{e}_a \times & \lambda \mathbf{e}_a \times & -\lambda \mathbf{e}_a \times \mathbf{f}_b \times & -\mathbf{d} \times \mathbf{e}_a \\ \mathbf{0} & -\lambda \mathbf{e}_a \times & \mathbf{0} & \mathbf{0} & \mathbf{e}_a \\ \mathbf{0} & -\lambda \mathbf{f}_b \times \mathbf{e}_a \times & \mathbf{0} & \lambda \mathbf{e}_a \times \mathbf{f}_b \times & \mathbf{f}_b \times \mathbf{e}_a \\ -\mathbf{e}_a^T & -(\mathbf{d} \times \mathbf{e}_a)^T & \mathbf{e}_a^T & (\mathbf{f}_b \times \mathbf{e}_a)^T & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \delta\mathbf{x}_a \\ \boldsymbol{\theta}_{a\delta} \\ \delta\mathbf{x}_b \\ \boldsymbol{\theta}_{b\delta} \\ \delta\lambda \end{array} \right\} \quad (9.74)$$

The constraint derivative is

$$\dot{\phi} = \mathbf{e}_a^T (\dot{\mathbf{x}}_b + \boldsymbol{\omega}_b \times \mathbf{f}_b - \dot{\mathbf{x}}_a) + \mathbf{d}^T (\boldsymbol{\omega}_a \times \mathbf{e}_a) - \dot{v} = 0, \quad (9.75)$$

or

$$\dot{\phi} = \mathbf{e}_a^T \dot{\mathbf{d}} + \mathbf{d}^T (\boldsymbol{\omega}_a \times \mathbf{e}_a) - \dot{v} = 0, \quad (9.76)$$

with $\dot{\mathbf{d}} = \dot{\mathbf{x}}_b + \boldsymbol{\omega}_b \times \mathbf{f}_b - \dot{\mathbf{x}}_a$. Its linearization yields

$$\begin{aligned} \delta\dot{\phi} = & \mathbf{e}_a^T \delta\dot{\mathbf{x}}_b - \mathbf{e}_a^T \delta\dot{\mathbf{x}}_a + (\mathbf{f}_b \times \mathbf{e}_a)^T \delta\boldsymbol{\omega}_b - (\mathbf{d} \times \mathbf{e}_a)^T \delta\boldsymbol{\omega}_a \\ & + (\boldsymbol{\omega}_a \times \mathbf{e}_a)^T \delta\mathbf{x}_b - (\boldsymbol{\omega}_a \times \mathbf{e}_a)^T \delta\mathbf{x}_a \\ & + (\mathbf{f}_b \times \mathbf{e}_a \times (\boldsymbol{\omega}_b - \boldsymbol{\omega}_a))^T \boldsymbol{\theta}_{b\delta} + \left(\mathbf{e}_a \times (\dot{\mathbf{d}} - \boldsymbol{\omega}_a \times \mathbf{d}) \right)^T \boldsymbol{\theta}_{a\delta} \end{aligned} \quad (9.77)$$

9.1.11 Imposed Displacement Pin

The imposed displacement pin joint (`ImposedDisplacementPinJoint`) is the “pinned” variant of the imposed displacement joint (`ImposedDisplacementJoint`) discussed in Section 9.1.10.

In this case, the direction e is referred to the absolute frame, so it does no longer depend on the state of the problem.

Variants. There exists a version that imposes the relative displacement of two nodes along a direction that is fixed with respect to one of them (`ImposedDisplacementJoint`), discussed in Section 9.1.10.

Files. It is implemented in files

`mbdyn/struct/impdisp.h`

`mbdyn/struct/impdisp.cc`

The constraint equation is

$$\phi = \mathbf{e}^T (\mathbf{x} + \mathbf{f} - \mathbf{x}_0) - v = 0, \quad (9.78)$$

or

$$\phi = \mathbf{e}^T \mathbf{d} - v = 0, \quad (9.79)$$

where $\mathbf{d} = \mathbf{x} + \mathbf{f} - \mathbf{x}_0$. The perturbation of the constraint equation yields

$$\delta\phi = \mathbf{e}^T (\delta\mathbf{x}_b - \mathbf{f}_b \times \boldsymbol{\theta}_{b\delta}). \quad (9.80)$$

The contribution of the constraint to the equilibrium equations is

$$\left\{ \begin{array}{c} \mathbf{F} \\ \mathbf{C} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{e}\lambda \\ \mathbf{f} \times \mathbf{e}\lambda \end{array} \right\} \quad (9.81)$$

where λ is the reaction force (a scalar).

The contribution of forces and moments and of the constraint equation to the Jacobian matrix is

$$\left[\begin{array}{ccc} \mathbf{0} & \mathbf{0} & \mathbf{e} \\ \mathbf{0} & \lambda\mathbf{e} \times \mathbf{f} \times & \mathbf{f} \times \mathbf{e} \\ \mathbf{e}^T & (\mathbf{f} \times \mathbf{e})^T & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \delta\mathbf{x} \\ \boldsymbol{\theta}_\delta \\ \delta\lambda \end{array} \right\} \quad (9.82)$$

The constraint derivative is

$$\dot{\phi} = \mathbf{e}^T (\dot{\mathbf{x}} + \boldsymbol{\omega} \times \mathbf{f}) - \dot{v} = 0, \quad (9.83)$$

or

$$\dot{\phi} = \mathbf{e}^T \dot{\mathbf{d}} - \dot{v} = 0, \quad (9.84)$$

Its linearization yields

$$\left[\begin{array}{cccccc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{e} & \mathbf{0} \\ \mathbf{0} & \lambda\mathbf{e} \times \mathbf{f} \times & \mathbf{0} & \mathbf{0} & \mathbf{f} \times \mathbf{e} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{e} \\ \mathbf{0} & \mu\mathbf{e} \times \mathbf{f} \times & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{f} \times \mathbf{e} \\ \mathbf{e}^T & (\mathbf{f} \times \mathbf{e})^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -(\mathbf{f} \times \boldsymbol{\omega} \times \mathbf{e})^T & \mathbf{e}^T & (\mathbf{f} \times \mathbf{e})^T & \mathbf{0} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \delta\mathbf{x} \\ \boldsymbol{\theta}_\delta \\ \delta\dot{\mathbf{x}} \\ \delta\boldsymbol{\omega} \\ \delta\lambda \\ \delta\mu \end{array} \right\} \quad (9.85)$$

9.1.12 Total Joint

This joint (`TotalJoint`) forces two nodes to assume an imposed relative position and orientation about selected axes. Both imposed position and orientation are specified in the reference frame of node 1. The joint consists in 6 equations, 3 for positions and 3 for orientation, that can be selectively turned on or off to enforce a constraint on that DOF.

Variants. There exists a pinned version of this joint that imposes the absolute position and orientation of a node (`TotalPinJoint`). It is discussed in Section 9.1.13.

Files. It is implemented in files

`mbdyn/struct/totalj.h`

`mbdyn/struct/totalj.cc`

Definitions. The position of the nodes with respect to the global reference frame are \mathbf{x}_1 and \mathbf{x}_2 . The matrices that describe the orientation of the nodes with respect to the global reference frame are \mathbf{R}_1 and \mathbf{R}_2 . An offset is allowed for the location of the points where the relative position is constrained; the offsets $\tilde{\mathbf{f}}_1$ and $\tilde{\mathbf{f}}_2$ are rigidly connected to the nodes, so in the global reference frame their expression is

$$\mathbf{f}_1 = \mathbf{R}_1 \tilde{\mathbf{f}}_1 \quad (9.86a)$$

$$\mathbf{f}_2 = \mathbf{R}_2 \tilde{\mathbf{f}}_2. \quad (9.86b)$$

The position constraint may be expressed in a reference frame that is rigidly connected to node 1, $\tilde{\mathbf{R}}_{1h}$, whose orientation in the global reference frame is

$$\mathbf{R}_{1h} = \mathbf{R}_1 \tilde{\mathbf{R}}_{1h}. \quad (9.87)$$

Similarly, the orientation constraint may be expressed in a reference frame that is rigidly connected to both nodes by matrices $\tilde{\mathbf{R}}_{1hr}$ and $\tilde{\mathbf{R}}_{2hr}$, whose orientation in the global reference frame is

$$\mathbf{R}_{1hr} = \mathbf{R}_1 \tilde{\mathbf{R}}_{1hr} \quad (9.88a)$$

$$\mathbf{R}_{2hr} = \mathbf{R}_2 \tilde{\mathbf{R}}_{2hr}. \quad (9.88b)$$

The relative position and orientation may refer to different local reference frames to increase the versatility of the element.

It is essential to notice that the differentiation of the constraint orientation matrices does not affect the relative orientation between the constraint and each node, namely matrices $\tilde{\mathbf{R}}_{1hr}$ and $\tilde{\mathbf{R}}_{2hr}$, which are constant. As a consequence,

$$\begin{aligned} \delta \mathbf{R}_{1hr} &= \delta \mathbf{R}_1 \tilde{\mathbf{R}}_{1hr} \\ &= \boldsymbol{\theta}_{1\delta} \times \mathbf{R}_1 \tilde{\mathbf{R}}_{1hr} \\ &= \boldsymbol{\theta}_{1\delta} \times \mathbf{R}_{1hr} \end{aligned} \quad (9.89a)$$

$$\delta \mathbf{R}_{2hr} = \boldsymbol{\theta}_{2\delta} \times \mathbf{R}_{2hr}, \quad (9.89b)$$

and

$$\dot{\mathbf{R}}_{1hr} = \boldsymbol{\omega}_1 \times \mathbf{R}_{1hr} \quad (9.90)$$

$$\dot{\mathbf{R}}_{2hr} = \boldsymbol{\omega}_2 \times \mathbf{R}_{2hr}. \quad (9.91)$$

The same is true for the relative position orientation matrix, \mathbf{R}_{1h} .

Orientation Constraint. The relative orientation is expressed by matrix

$$\mathbf{R}_{\text{rel}} = \mathbf{R}_{1hr}^T \mathbf{R}_{2hr} \quad (9.92)$$

The desired relative orientation can be expressed by means of a rotation vector $\boldsymbol{\theta}_0$; the corresponding rotation matrix is

$$\mathbf{R}_0 = \exp(\boldsymbol{\theta}_0 \times). \quad (9.93)$$

This operation can always be performed with no ambiguity, regardless of the amplitude of $\boldsymbol{\theta}_0$. However, the same is not true for the inverse operation, namely

$$\boldsymbol{\theta}_0 = \text{ax}(\exp^{-1}(\mathbf{R}_0)), \quad (9.94)$$

which can only yield the minimal $\boldsymbol{\theta}_0$ corresponding to \mathbf{R}_0 , with the indetermination corresponding to an arbitrary amount of complete revolutions.

The equality

$$\boldsymbol{\theta} = \boldsymbol{\theta}_0 \quad (9.95)$$

implies that

$$\mathbf{R}_{\text{rel}} = \mathbf{R}_0. \quad (9.96)$$

Equation (9.96) can be rewritten as

$$\mathbf{R}_{\text{rel}} \mathbf{R}_0^T = \mathbf{I}, \quad (9.97)$$

and thus Eq. (9.95), after defining

$$\mathbf{R}^\delta = \mathbf{R}_{\text{rel}} \mathbf{R}_0^T, \quad (9.98)$$

is equivalent to

$$\text{ax}(\exp^{-1}(\mathbf{R}^\delta)) = \mathbf{0}. \quad (9.99)$$

In general, the difference between the desired rotation $\boldsymbol{\theta}_0$ and the current relative rotation vector between nodes 1 and 2,

$$\boldsymbol{\theta} = \text{ax}(\exp^{-1}(\mathbf{R}_{1hr}^T \mathbf{R}_{2hr})) \quad (9.100)$$

through the previous redefinition can be expressed as

$$\boldsymbol{\theta}^\delta = \text{ax}(\exp^{-1}(\mathbf{R}_{1hr}^T \mathbf{R}_{2hr} \mathbf{R}_0^T)) \quad (9.101)$$

and is equivalent to

$$\boldsymbol{\theta}^\delta = \boldsymbol{\theta} - \boldsymbol{\theta}_0 \quad (9.102)$$

when $\boldsymbol{\theta}^\delta = \mathbf{0}$.

Position Constraint. The desired relative position is imposed by means of a vector \mathbf{x}_0 defined in the relative position constraint reference frame of node 1. The relative position is represented by vector \mathbf{x} , defined as

$$\mathbf{x} = \mathbf{R}_{1h}^T (\mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1 - \mathbf{f}_1); \quad (9.103)$$

the constraint equations for the relative position are:

$$\mathbf{x} = \mathbf{x}_0. \quad (9.104)$$

In analogy with the definition of $\boldsymbol{\theta}^\delta$, a vector \mathbf{x}^δ can be used to express the constraint on the position DOFs,

$$\mathbf{x}^\delta = \mathbf{x} - \mathbf{x}_0. \quad (9.105)$$

Constraint Equations. The resulting constraint equations are:

$$\boldsymbol{x}^\delta = \mathbf{0} \quad (9.106a)$$

$$\boldsymbol{\theta}^\delta = \mathbf{0}. \quad (9.106b)$$

Any component of \boldsymbol{x}^δ and $\boldsymbol{\theta}^\delta$ can be selectively set to zero to enforce a constraint on that degree of freedom. The axis about which the constraint is applied can be arbitrarily defined with respect to the node orientation by appropriately setting the constant relative orientation matrix of each node, both for position ($\tilde{\mathbf{R}}_{1h}$) and for orientation ($\tilde{\mathbf{R}}_{1hr}$ and $\tilde{\mathbf{R}}_{2hr}$). For imposed rotations, an arbitrary amplitude of the rotation about the constraint axis can be imposed by means of the components of vector $\boldsymbol{\theta}_0$. The same is true for positions, by means of $\tilde{\mathbf{v}}$.

Alternatively, equations related to any of the components of \boldsymbol{x}^δ and $\boldsymbol{\theta}^\delta$ can be omitted, thus omitting the constraint along or about the corresponding axis. The corresponding value of \boldsymbol{x}^δ and $\boldsymbol{\theta}^\delta$ could be used to apply moments based on the relative position/orientation of the nodes, e.g. linear/rotational springs or dampers. In this case, the singularity problem on rotations is back: the norm of the relative rotation must not exceed π , i.e. $|\boldsymbol{\theta}^\delta| < \pi$. Note that the relative rotation is computed with respect to $\boldsymbol{\theta}_0$, so an anelastic (or imposed) rotation can be easily applied to the spring, resulting in an imposed rotation with stiffness.

Linearization. First, the perturbation of \boldsymbol{x}^δ ,

$$\delta\boldsymbol{x}^\delta = \mathbf{R}_{1h}^T (\delta\boldsymbol{x}_2 - \mathbf{b}_2 \times \boldsymbol{\theta}_{2\delta} - \delta\boldsymbol{x}_1 + \mathbf{b}_1 \times \boldsymbol{\theta}_{1\delta}) \quad (9.107)$$

is computed, where

$$\mathbf{b}_1 = \boldsymbol{x}_2 + \mathbf{f}_2 - \boldsymbol{x}_1 \quad (9.108a)$$

$$\mathbf{b}_2 = \mathbf{f}_2. \quad (9.108b)$$

Note that, since the constraint equations are defined by setting to zero any components of the vector \boldsymbol{x}^δ , the related Jacobian matrix contribution is obtained by selecting the corresponding columns of matrix \mathbf{R}_{1h} .

Then, the perturbation of $\boldsymbol{\theta}^\delta$ results from

$$\begin{aligned} \boldsymbol{\theta}_\delta^\delta \times &= \delta\mathbf{R}^\delta \mathbf{R}^{\delta T} \\ &= \mathbf{R}_{1hr}^T \boldsymbol{\theta}_{1\delta} \times^T \mathbf{R}_{1hr} + \mathbf{R}_{1hr}^T \boldsymbol{\theta}_{2\delta} \times \mathbf{R}_{1hr} \\ &= \mathbf{R}_{1hr}^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) \times \mathbf{R}_{1hr}, \end{aligned} \quad (9.109)$$

which implies

$$\boldsymbol{\theta}_\delta^\delta = \mathbf{R}_{1hr}^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}). \quad (9.110)$$

Note that, since the constraint equations are defined by setting to zero any components of the vector $\boldsymbol{\theta}^\delta$, the related Jacobian matrix contribution is obtained by selecting the corresponding columns of matrix \mathbf{R}_{1hr} .

After calling $\tilde{\boldsymbol{\lambda}}_x$ and $\tilde{\boldsymbol{\lambda}}_\theta$ the Lagrange multipliers respectively related to the relative position and orientation constraints, and noticing that in the global reference frame they become

$$\boldsymbol{\lambda}_x = \mathbf{R}_{1h} \tilde{\boldsymbol{\lambda}}_x \quad (9.111a)$$

$$\boldsymbol{\lambda}_\theta = \mathbf{R}_{1hr} \tilde{\boldsymbol{\lambda}}_\theta, \quad (9.111b)$$

the contribution of the constraint reactions to the equilibrium equations of the participating nodes is

$$\mathbf{F}_1 = -\boldsymbol{\lambda}_{\mathbf{x}} \quad (9.112a)$$

$$\mathbf{C}_1 = -\mathbf{b}_1 \times \boldsymbol{\lambda}_{\mathbf{x}} - \boldsymbol{\lambda}_{\theta} \quad (9.112b)$$

$$\mathbf{F}_2 = \boldsymbol{\lambda}_{\mathbf{x}} \quad (9.112c)$$

$$\mathbf{C}_2 = \mathbf{b}_2 \times \boldsymbol{\lambda}_{\mathbf{x}} + \boldsymbol{\lambda}_{\theta}. \quad (9.112d)$$

Due to the simplifications related to the updated-updated approach, the contribution of the constraint equations and of the equilibrium to the Jacobian matrix is:

$$\left[\begin{array}{ccc|cc} \mathbf{0} & \boldsymbol{\lambda}_{\mathbf{x}} \times & \mathbf{0} & \mathbf{0} & -\mathbf{R}_{1h} & \mathbf{0} \\ -\boldsymbol{\lambda}_{\mathbf{x}} \times & \mathbf{b}_1 \times \boldsymbol{\lambda}_{\mathbf{x}} \times + \boldsymbol{\lambda}_{\theta} \times & \boldsymbol{\lambda}_{\mathbf{x}} \times & -\boldsymbol{\lambda}_{\mathbf{x}} \times \mathbf{b}_2 \times & -\mathbf{b}_1 \times \mathbf{R}_{1h} & -\mathbf{R}_{1hr} \\ \mathbf{0} & -\boldsymbol{\lambda}_{\mathbf{x}} \times & \mathbf{0} & \mathbf{0} & \mathbf{R}_{1h} & \mathbf{0} \\ \mathbf{0} & -\mathbf{b}_2 \times \boldsymbol{\lambda}_{\mathbf{x}} \times - \boldsymbol{\lambda}_{\theta} \times & \mathbf{0} & \boldsymbol{\lambda}_{\mathbf{x}} \times \mathbf{b}_2 \times & \mathbf{b}_2 \times \mathbf{R}_{1h} & \mathbf{R}_{1hr} \\ -\mathbf{R}_{1h}^T & \mathbf{R}_{1h}^T \mathbf{b}_1 \times & \mathbf{R}_{1h}^T & -\mathbf{R}_{1h}^T \mathbf{b}_2 \times & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{R}_{1hr}^T & \mathbf{0} & \mathbf{R}_{1hr}^T & \mathbf{0} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_1 \\ \theta_{1\delta} \\ \delta \mathbf{x}_2 \\ \theta_{2\delta} \\ \delta \boldsymbol{\lambda}_{\mathbf{x}} \\ \delta \boldsymbol{\lambda}_{\theta} \end{array} \right\}$$

$$= \left\{ \begin{array}{l} \boldsymbol{\lambda}_{\mathbf{x}} \\ \mathbf{b}_1 \times \boldsymbol{\lambda}_{\mathbf{x}} + \boldsymbol{\lambda}_{\theta} \\ -\boldsymbol{\lambda}_{\mathbf{x}} \\ -\mathbf{b}_2 \times \boldsymbol{\lambda}_{\mathbf{x}} - \boldsymbol{\lambda}_{\theta} \\ -\mathbf{x}^\delta \\ -\theta^\delta \end{array} \right\}. \quad (9.113)$$

The updated-updated form is obtained by simply replacing $\theta_{1\delta}$ and $\theta_{2\delta}$ with $\delta \mathbf{g}_1$ and $\delta \mathbf{g}_2$, respectively.

Summary

- each component of the relative position/orientation can be subjected to separate constraint conditions;
- an ideal constraint results from activating the constraint equation on that degree of freedom and setting to zero, or to any desired, time dependent value the relative position/orientation of that component;
- a deformable constraint results from deactivating the constraint equation on that degree of freedom, possibly adding a constitutive law based on that error and possibly on its derivative (currently, this requires a separate instance of a deformable element); in this case, the norm of the amplitude of the relative orientation is limited to π ;
- the imposed relative orientation goes in matrix \mathbf{R}_0 by means of a rotation vector $\boldsymbol{\theta}_0$ which can be time dependent; there is no limitation on the amplitude of that rotation;
- friction can be added as well on those components that are not constrained;
- an interesting option, left as a future development, consists in leaving all the constraint equations always in place, and activate/deactivate them based on some trigger.

Future Development: Velocity/Acceleration

The time derivative of vector \mathbf{x}^δ yields

$$\mathbf{v}^\delta = \dot{\mathbf{x}} - \mathbf{v}_0 = \mathbf{R}_{1h}^T (\dot{\mathbf{x}}_2 + \boldsymbol{\omega}_2 \times \mathbf{f}_2 - \dot{\mathbf{x}}_1 - \boldsymbol{\omega}_1 \times (\mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1)) - \mathbf{v}_0. \quad (9.114)$$

Its linearization yields

$$\begin{aligned}
\delta \mathbf{v}^\delta &= \delta \dot{\mathbf{x}} \\
&= \mathbf{R}_{1h}^T \delta \dot{\mathbf{x}}_2 \\
&\quad - \mathbf{R}_{1h}^T \delta \dot{\mathbf{x}}_1 \\
&\quad - \mathbf{R}_{1h}^T \mathbf{f}_2 \times \delta \boldsymbol{\omega}_2 \\
&\quad + \mathbf{R}_{1h}^T (\mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1) \times \delta \boldsymbol{\omega}_1 \\
&\quad - \mathbf{R}_{1h}^T \boldsymbol{\omega}_1 \times \delta \mathbf{x}_2 \\
&\quad + \mathbf{R}_{1h}^T \boldsymbol{\omega}_1 \times \delta \mathbf{x}_1 \\
&\quad - \mathbf{R}_{1h}^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} \\
&\quad + \mathbf{R}_{1h}^T (\dot{\mathbf{x}}_2 + \boldsymbol{\omega}_2 \times \mathbf{f}_2 - \dot{\mathbf{x}}_1 - \boldsymbol{\omega}_1 \times (\mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1)) \times \boldsymbol{\theta}_{1\delta} \\
&\stackrel{\text{uu}}{=} \mathbf{R}_{1h}^T \delta \dot{\mathbf{x}}_2 \\
&\quad - \mathbf{R}_{1h}^T \delta \dot{\mathbf{x}}_1 \\
&\quad - \mathbf{R}_{1h}^T \mathbf{f}_2 \times \delta \dot{\mathbf{g}}_2 \\
&\quad + \mathbf{R}_{1h}^T (\mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1) \times \delta \dot{\mathbf{g}}_1 \\
&\quad - \mathbf{R}_{1h}^T \boldsymbol{\omega}_1 \times \delta \mathbf{x}_2 \\
&\quad + \mathbf{R}_{1h}^T \boldsymbol{\omega}_1 \times \delta \mathbf{x}_1 \\
&\quad + \mathbf{R}_{1h}^T ((\mathbf{f}_2 \times \boldsymbol{\omega}_2) \times + \boldsymbol{\omega}_1 \times \mathbf{f}_2 \times) \boldsymbol{\theta}_{2\delta} \\
&\quad + \mathbf{R}_{1h}^T ((\dot{\mathbf{x}}_2 + \boldsymbol{\omega}_2 \times \mathbf{f}_2 - \dot{\mathbf{x}}_1) \times - \boldsymbol{\omega}_1 \times (\mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1) \times) \boldsymbol{\theta}_{1\delta}. \tag{9.115}
\end{aligned}$$

The time derivative of matrix \mathbf{R}^δ yields the constraint equation on the angular velocity

$$\begin{aligned}
\boldsymbol{\omega}^\delta \times &= \dot{\mathbf{R}}^\delta \mathbf{R}^{\delta T} \\
&= \mathbf{R}_{1hr}^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \mathbf{R}_{1hr} - \mathbf{R}^\delta \boldsymbol{\omega}_0 \times \mathbf{R}^{\delta T}, \tag{9.116}
\end{aligned}$$

which results in

$$\boldsymbol{\omega}^\delta = \mathbf{R}_{1hr}^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) - \mathbf{R}^\delta \boldsymbol{\omega}_0, \tag{9.117}$$

where $\boldsymbol{\omega}_0$ is the derivative of the imposed orientation,

$$\boldsymbol{\omega}_0 = \dot{\boldsymbol{\theta}}_0. \tag{9.118}$$

Note that if the angular velocity constraint is the derivative of an orientation constraint, $\mathbf{R}^\delta = \mathbf{I}$, so, by setting $\boldsymbol{\omega}^\delta = \mathbf{0}$, Equation (9.117) becomes

$$\boldsymbol{\omega}_0 = \mathbf{R}_{1hr}^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1). \tag{9.119}$$

Otherwise, Equation (9.117) represents a non-holonomic constraint. In this case, \mathbf{R}^δ is again the identity matrix, although resulting from an unknown \mathbf{R}_0 , which is the result of integrating the dynamics of the problem with the non-holonomic constraint represented by Equation (9.119). In fact,

$$\mathbf{R}_0 = \mathbf{R}_{1hr}^T \mathbf{R}_{2hr} \tag{9.120}$$

is the definition of the relative orientation resulting from imposing the constraint of Equation (9.119).

The perturbation of Equation (9.119) results in

$$\begin{aligned}\delta(\mathbf{R}_{1hr}^T(\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1)) &= \mathbf{R}_{1hr}^T(\delta\boldsymbol{\omega}_2 - \delta\boldsymbol{\omega}_1 + (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \boldsymbol{\theta}_{1\delta}) \\ &\stackrel{\text{uu}}{=} \mathbf{R}_{1hr}^T(\delta\dot{\boldsymbol{g}}_2 - \delta\dot{\boldsymbol{g}}_1 - \boldsymbol{\omega}_2 \times \delta\boldsymbol{g}_2 + \boldsymbol{\omega}_2 \times \delta\boldsymbol{g}_1)\end{aligned}\quad (9.121)$$

Further time differentiation of the constraint equation results in

$$\dot{\boldsymbol{\omega}}^\delta = \mathbf{R}_{1hr}^T(\dot{\boldsymbol{\omega}}_2 - \dot{\boldsymbol{\omega}}_1 - \boldsymbol{\omega}_1 \times \boldsymbol{\omega}_2) - \boldsymbol{\omega}^\delta \times \mathbf{R}^\delta \boldsymbol{\omega}_0 - \mathbf{R}^\delta \dot{\boldsymbol{\omega}}_0. \quad (9.122)$$

Similarly, if the angular acceleration constraint is the derivative of an orientation or an angular velocity constraint, then $\mathbf{R}^\delta = \mathbf{I}$, $\boldsymbol{\omega}^\delta = \mathbf{0}$, respectively, and, by setting $\dot{\boldsymbol{\omega}}^\delta = \mathbf{0}$, Equation (9.122) becomes

$$\dot{\boldsymbol{\omega}}_0 = \mathbf{R}_{1hr}^T(\dot{\boldsymbol{\omega}}_2 - \dot{\boldsymbol{\omega}}_1 - \boldsymbol{\omega}_1 \times \boldsymbol{\omega}_2). \quad (9.123)$$

9.1.13 Total Pin Joint

This joint (`TotalPinJoint`) forces a node to assume an imposed absolute position and orientation about selected axes. Both imposed position and orientation are specified in the global reference frame, optionally re-oriented by a constant orientation. The joint consists in 6 equations, 3 for positions and 3 for orientation, that can be selectively turned on or off to enforce a constraint on that DOF.

Variants. This joint is a variant of the `TotalJoint`, described in Section 9.1.12.

Files. It is implemented in files
`mbdyn/struct/totalj.h`
`mbdyn/struct/totalj.cc`

Definitions. The position of the node with respect to the global reference frame is \mathbf{x}_n , while the absolute position of the constraint is \mathbf{x}_c . The matrix that expresses the orientation of the node with respect to the global reference frame is \mathbf{R}_n . An offset is allowed for the location of the points where the absolute position is constrained; the offset $\tilde{\mathbf{f}}_n$ is rigidly connected to the node, so in the global reference frame its expression is

$$\mathbf{f}_n = \mathbf{R}_n \tilde{\mathbf{f}}_n \quad (9.124)$$

The position constraint may be expressed in a reference frame, whose orientation in the global reference frame is \mathbf{R}_{ch} . Later on, it will occasionally be convenient to express the absolute position of the constraint, \mathbf{x}_c , in the reference frame \mathbf{R}_{ch} , namely

$$\tilde{\mathbf{x}}_c = \mathbf{R}_{ch}^T \mathbf{x}_c. \quad (9.125)$$

Similarly, the orientation constraint may be expressed in a reference frame that is rigidly connected to the node by matrix $\tilde{\mathbf{R}}_{nhr}$, whose orientation in the global reference frame is

$$\mathbf{R}_{nhr} = \mathbf{R}_n \tilde{\mathbf{R}}_{nhr}, \quad (9.126)$$

while the absolute orientation is described by the constant matrix \mathbf{R}_{chr} . The absolute position and orientation may refer to different local reference frames to increase the versatility of the element.

It is essential to notice that the differentiation of the constraint orientation matrices does not affect the relative orientation between the constraint and the node, namely matrix $\tilde{\mathbf{R}}_{nhr}$, which is constant. As a consequence,

$$\begin{aligned}\delta \mathbf{R}_{nhr} &= \delta \mathbf{R}_n \tilde{\mathbf{R}}_{nhr} \\ &= \boldsymbol{\theta}_{n\delta} \times \mathbf{R}_n \tilde{\mathbf{R}}_{nhr} \\ &= \boldsymbol{\theta}_{n\delta} \times \mathbf{R}_{nhr},\end{aligned}\tag{9.127}$$

and

$$\dot{\mathbf{R}}_{nhr} = \boldsymbol{\omega}_n \times \mathbf{R}_{nhr}.\tag{9.128}$$

The same is true for the relative position orientation matrix, \mathbf{R}_{nh} .

Orientation Constraint. The matrix that expresses the relative orientation is

$$\mathbf{R}_{\text{rel}} = \mathbf{R}_{chr}^T \mathbf{R}_{nhr}\tag{9.129}$$

The desired relative orientation is expressed in analogy with the **TotalJoint**.

In general, the difference between the desired rotation $\boldsymbol{\theta}_0$ and the current absolute orientation of the node,

$$\boldsymbol{\theta} = \text{ax}(\exp^{-1}(\mathbf{R}_{chr}^T \mathbf{R}_{nhr}))\tag{9.130}$$

through the redefinition of Eq. (9.98) can be expressed as

$$\boldsymbol{\theta}^\delta = \text{ax}(\exp^{-1}(\mathbf{R}_{chr}^T \mathbf{R}_{nhr} \mathbf{R}_0^T))\tag{9.131}$$

and is equivalent to

$$\boldsymbol{\theta}^\delta = \boldsymbol{\theta} - \boldsymbol{\theta}_0\tag{9.132}$$

when $\boldsymbol{\theta}^\delta = \mathbf{0}$.

Position Constraint. The desired absolute position is imposed by means of a vector \mathbf{x}_0 defined in the absolute position constraint reference frame \mathbf{R}_{ch} . The absolute position is represented by vector \mathbf{x} , defined as

$$\begin{aligned}\mathbf{x} &= \mathbf{R}_{ch}^T (\mathbf{x}_n + \mathbf{f}_n - \mathbf{x}_c) \\ &= \mathbf{R}_{ch}^T (\mathbf{x}_n + \mathbf{f}_n) - \tilde{\mathbf{x}}_c;\end{aligned}\tag{9.133}$$

the constraint equations for the relative position are:

$$\mathbf{x} = \mathbf{x}_0.\tag{9.134}$$

In analogy with the definition of $\boldsymbol{\theta}^\delta$, a vector \mathbf{x}^δ can be used to express the constraint on the position DOFs,

$$\mathbf{x}^\delta = \mathbf{x} - \mathbf{x}_0.\tag{9.135}$$

Constraint Equations. The resulting constraint equations are identical to Eqs. (9.106a, 9.106b), related to the **TotalJoint**, as illustrated at page 66; the same considerations apply.

Linearization. First, the perturbation of \mathbf{x}^δ ,

$$\delta\mathbf{x}^\delta = \mathbf{R}_{ch}^T (\delta\mathbf{x}_n - \mathbf{f}_n \times \boldsymbol{\theta}_{n\delta}) \quad (9.136)$$

is computed. Note that, since the constraint equations are defined by setting to zero any components of the vector \mathbf{x}^δ , the related Jacobian matrix contribution is obtained by selecting the corresponding columns of matrix \mathbf{R}_{ch} .

Then, the perturbation of $\boldsymbol{\theta}^\delta$ results from

$$\begin{aligned} \boldsymbol{\theta}_\delta^\delta \times &= \delta\mathbf{R}^\delta \mathbf{R}^{\delta T} \\ &= \mathbf{R}_{chr}^T \boldsymbol{\theta}_{n\delta} \times \mathbf{R}_{chr}, \end{aligned} \quad (9.137)$$

which implies

$$\boldsymbol{\theta}_\delta^\delta = \mathbf{R}_{chr}^T \boldsymbol{\theta}_{n\delta}. \quad (9.138)$$

Note that, since the constraint equations are defined by setting to zero any components of the vector $\boldsymbol{\theta}^\delta$, the related Jacobian matrix contribution is obtained by selecting the corresponding columns of matrix \mathbf{R}_{chr} .

After calling $\tilde{\boldsymbol{\lambda}}_x$ and $\tilde{\boldsymbol{\lambda}}_\theta$ the Lagrange multipliers respectively related to the relative position and orientation constraints, and noticing that in the global reference frame they become

$$\boldsymbol{\lambda}_x = \mathbf{R}_{ch} \tilde{\boldsymbol{\lambda}}_x \quad (9.139a)$$

$$\boldsymbol{\lambda}_\theta = \mathbf{R}_{chr} \tilde{\boldsymbol{\lambda}}_\theta, \quad (9.139b)$$

the contribution of the constraint reactions to the equilibrium equations of the participating nodes is

$$\mathbf{F}_n = \boldsymbol{\lambda}_x \quad (9.140a)$$

$$\mathbf{C}_n = \mathbf{f}_n \times \boldsymbol{\lambda}_x + \boldsymbol{\lambda}_\theta. \quad (9.140b)$$

Due to the simplifications related to the updated-updated approach, the contribution of the constraint equations and of the equilibrium to the Jacobian matrix is:

$$\left[\begin{array}{cc|cc} \mathbf{0} & \mathbf{0} & \mathbf{R}_{ch} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda}_x \times \mathbf{f}_n \times & \mathbf{f}_n \times \mathbf{R}_{ch} & \mathbf{R}_{chr} \\ \hline \mathbf{R}_{ch}^T & -\mathbf{R}_{ch}^T \mathbf{f}_n \times & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{chr}^T & \mathbf{0} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \delta\mathbf{x}_n \\ \boldsymbol{\theta}_{n\delta} \\ \delta\tilde{\boldsymbol{\lambda}}_x \\ \delta\tilde{\boldsymbol{\lambda}}_\theta \end{array} \right\} = \left\{ \begin{array}{c} -\boldsymbol{\lambda}_x \\ -\mathbf{f}_n \times \boldsymbol{\lambda}_x - \boldsymbol{\lambda}_\theta \\ -\mathbf{x}^\delta \\ -\boldsymbol{\theta}^\delta \end{array} \right\}. \quad (9.141)$$

The updated-updated form is obtained by simply replacing $\boldsymbol{\theta}_{n\delta}$ with $\delta\mathbf{g}_n$.

Stabilized Form

Equations.

$$\mathbf{R}_{ch} \boldsymbol{\mu}_x = \mathbf{0} \quad (9.142a)$$

$$\mathbf{f}_n \times \mathbf{R}_{ch} \boldsymbol{\mu}_x + \mathbf{R}_{chr} \boldsymbol{\mu}_\theta = \mathbf{0} \quad (9.142b)$$

$$\mathbf{R}_{ch} \boldsymbol{\lambda}_x = \mathbf{0} \quad (9.142c)$$

$$\begin{aligned} -\mathbf{f}_n \times \boldsymbol{\omega}_n \times \mathbf{R}_{ch} \boldsymbol{\mu}_x - \mathbf{R}_{chr} (\mathbf{R}^\delta \boldsymbol{\omega}_0) \times \boldsymbol{\mu}_\theta \\ + \mathbf{f}_n \times \mathbf{R}_{ch} \boldsymbol{\lambda}_x + \mathbf{R}_{chr} \boldsymbol{\lambda}_\theta = \mathbf{0} \end{aligned} \quad (9.142d)$$

$$\mathbf{R}_{ch}^T (\mathbf{x}_n + \mathbf{f}_n) - \mathbf{x}_0 = \mathbf{0} \quad (9.142e)$$

$$\text{ax}(\exp^{-1}(\mathbf{R}^\delta)) = \mathbf{0} \quad (9.142f)$$

$$\mathbf{R}_{ch}^T (\dot{\mathbf{x}}_n + \boldsymbol{\omega}_n \times \mathbf{f}_n) - \dot{\mathbf{x}}_0 = \mathbf{0} \quad (9.142g)$$

$$\mathbf{R}_{chr}^T \boldsymbol{\omega}_n - \mathbf{R}^\delta \boldsymbol{\omega}_0 = \mathbf{0} \quad (9.142h)$$

Linearization.

$$\mathbf{R}_{ch}\delta\boldsymbol{\mu}_x = \mathbf{0} \quad (9.143a)$$

$$-(\mathbf{f}_n \times \mathbf{R}_{ch}\boldsymbol{\mu}_x) \times \boldsymbol{\theta}_{n\delta} + \mathbf{f}_n \times \mathbf{R}_{ch}\delta\boldsymbol{\mu}_x + \mathbf{R}_{chr}\delta\boldsymbol{\mu}_\theta = \mathbf{0} \quad (9.143b)$$

$$\mathbf{R}_{ch}\delta\boldsymbol{\lambda}_x = \mathbf{0} \quad (9.143c)$$

$$(\mathbf{f}_n \times \boldsymbol{\omega}_n \times \mathbf{R}_{ch}\boldsymbol{\mu}_x) \times \boldsymbol{\theta}_{n\delta} + \mathbf{f}_n \times (\mathbf{R}_{ch}\boldsymbol{\mu}_x) \times \delta\boldsymbol{\omega}_n$$

$$-\mathbf{f}_n \times \boldsymbol{\omega}_n \times \mathbf{R}_{ch}\delta\boldsymbol{\mu}_x$$

$$-(\mathbf{R}_{chr}\boldsymbol{\mu}_\theta) \times (\mathbf{R}_{nhn}\mathbf{R}_0^T\boldsymbol{\omega}_0) \times \boldsymbol{\theta}_{n\delta} - \mathbf{R}_{chr}(\mathbf{R}^\delta\boldsymbol{\omega}_0) \times \delta\boldsymbol{\mu}_\theta \\ - (\mathbf{f}_n \times \mathbf{R}_{ch}\boldsymbol{\lambda}_x) \times \boldsymbol{\theta}_{n\delta} + \mathbf{f}_n \times \mathbf{R}_{ch}\delta\boldsymbol{\lambda}_x + \mathbf{R}_{chr}\delta\boldsymbol{\lambda}_\theta = \mathbf{0} \quad (9.143d)$$

$$\mathbf{R}_{ch}^T\delta\mathbf{x}_n - \mathbf{R}_{ch}^T\mathbf{f}_n \times \boldsymbol{\theta}_{n\delta} = \mathbf{0} \quad (9.143e)$$

$$\mathbf{R}_{chr}^T\boldsymbol{\theta}_{n\delta} = \mathbf{0} \quad (9.143f)$$

$$\mathbf{R}_{ch}^T\delta\dot{\mathbf{x}}_n - \mathbf{R}_{ch}^T\mathbf{f}_n \times \delta\boldsymbol{\omega}_n - \mathbf{R}_{ch}^T\boldsymbol{\omega}_n \times \mathbf{f}_n \times \boldsymbol{\theta}_{n\delta} = \mathbf{0} \quad (9.143g)$$

$$\mathbf{R}_{chr}^T\delta\boldsymbol{\omega}_n + \mathbf{R}_{chr}^T(\mathbf{R}_{nhn}\mathbf{R}_0^T\boldsymbol{\omega}_0) \times \boldsymbol{\theta}_{n\delta} = \mathbf{0} \quad (9.143h)$$

Summary

The same considerations formulated for the **TotalJoint** apply.

9.1.14 Gimbal

This element implements an ideal gimbal joint. It is discussed in [5]. A gimbal is a joint that allows the rotation between two nodes about two orthogonal axes. The angular velocity about the remaining axis is preserved regardless of the relative angle between the two nodes. It is essentially equivalent to two Cardano joints³ each of which accounts for half of the relative orientation.

Variants. This joint is only defined in the variant that constrains the relative rotation between two nodes. It may need to be combined with another joint that constrains the (relative) position of the related nodes, and with a deformable hinge to provide some stiffness or damping.

Files. It is implemented in files

`mbdyn/struct/gimbal.h`

`mbdyn/struct/gimbal.cc`

Definitions. The relative orientation between the nodes is made of three steps, described by

$$\mathbf{R}_{rel} = \mathbf{R}_a^T \mathbf{R}_b \quad (= \mathbf{R}_{ab}) \quad (9.144a)$$

$$= \exp(\vartheta \mathbf{e}_2 \times) \exp(\varphi \mathbf{e}_1 \times) \exp(\vartheta \mathbf{e}_2 \times) \quad (= \mathbf{R}(\vartheta, \varphi)) \quad (9.144b)$$

where

$$\exp(\vartheta \mathbf{e}_2 \times) = \mathbf{I} + \sin \vartheta \mathbf{e}_2 \times + (1 - \cos \vartheta) \mathbf{e}_2 \times \mathbf{e}_2 \times \quad (9.145a)$$

$$\exp(\varphi \mathbf{e}_1 \times) = \mathbf{I} + \sin \varphi \mathbf{e}_1 \times + (1 - \cos \varphi) \mathbf{e}_1 \times \mathbf{e}_1 \times \quad (9.145b)$$

and \mathbf{e}_i are the unit vectors in directions $i = 1, 2, 3$.

³Or “universal joints”; this is also known as a double Hooke joint.

This represents a sequence of mutually orthogonal rotations, symmetrical with respect to the mid of the central rotation, and thus consisting in a sequence of two Cardano joints in the “W” arrangement, which is homokinetic.

The constraint equation is obtained by equating the relative orientation matrices of Equations (9.144a, 9.144b):

$$\mathbf{R}_{ab} = \mathbf{R}(\vartheta, \varphi), \quad (9.146)$$

or, which is equivalent,

$$\mathbf{R}_{ab}\mathbf{R}(\vartheta, \varphi)^T = \mathbf{I}. \quad (9.147)$$

As a consequence, the rotation vector

$$\boldsymbol{\theta} = \text{ax} \left(\exp^{-1} \left(\mathbf{R}_{ab}\mathbf{R}(\vartheta, \varphi)^T \right) \right), \quad (9.148)$$

corresponding to the orientation matrix at the left-hand side of Equation (9.147), must vanish:

$$\boldsymbol{\theta} = \mathbf{0}. \quad (9.149)$$

Lagrange multipliers approach:

$$\begin{aligned} 0 &= \delta(\boldsymbol{\lambda}^T \boldsymbol{\theta}) \\ &= \delta\boldsymbol{\lambda}^T \boldsymbol{\theta} + \delta\boldsymbol{\theta}^T \boldsymbol{\lambda}, \end{aligned} \quad (9.150)$$

where

$$\delta\boldsymbol{\theta} = \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \boldsymbol{\theta}_\delta. \quad (9.151)$$

Note, that by definition $\boldsymbol{\theta}$ must be zero, so $\boldsymbol{\Gamma}(\boldsymbol{\theta}) \cong \mathbf{I}$. Without any simplification, Equation (9.150) results in

$$0 = \delta(\boldsymbol{\lambda}^T \boldsymbol{\theta}) \quad (9.152a)$$

$$= \delta\boldsymbol{\lambda}^T \boldsymbol{\theta} + \boldsymbol{\theta}_\delta^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \boldsymbol{\lambda}. \quad (9.152b)$$

To eliminate the $\boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T}$ term, the multipliers can be redefined as

$$\hat{\boldsymbol{\lambda}} = \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \boldsymbol{\lambda}. \quad (9.153)$$

The perturbation $\boldsymbol{\theta}_\delta$ results from

$$\begin{aligned} \boldsymbol{\theta}_\delta \times &= \delta \exp(\boldsymbol{\theta} \times) \exp(\boldsymbol{\theta} \times)^T \\ &= \delta \left(\mathbf{R}_a^T \mathbf{R}_b \exp(\vartheta \mathbf{e}_2 \times)^T \exp(\varphi \mathbf{e}_1 \times)^T \exp(\vartheta \mathbf{e}_2 \times)^T \right) \cdot \\ &\quad \left(\mathbf{R}_a^T \mathbf{R}_b \exp(\vartheta \mathbf{e}_2 \times)^T \exp(\varphi \mathbf{e}_1 \times)^T \exp(\vartheta \mathbf{e}_2 \times)^T \right)^T \\ &= \mathbf{R}_a^T (\boldsymbol{\theta}_{b\delta} - \boldsymbol{\theta}_{a\delta}) \times \mathbf{R}_a \\ &- \delta \vartheta \mathbf{R}_a^T \mathbf{R}_b \left(\left(\mathbf{I} + \exp(\vartheta \mathbf{e}_2 \times)^T \exp(\varphi \mathbf{e}_1 \times)^T \right) \mathbf{e}_2 \right) \times \mathbf{R}_b^T \mathbf{R}_a \\ &- \delta \varphi \mathbf{R}_a^T \mathbf{R}_b \left(\exp(\vartheta \mathbf{e}_2 \times)^T \mathbf{e}_1 \right) \times \mathbf{R}_b^T \mathbf{R}_a \end{aligned} \quad (9.154)$$

which yields

$$\boldsymbol{\theta}_\delta = \mathbf{R}_a^T (\boldsymbol{\theta}_{b\delta} - \boldsymbol{\theta}_{a\delta}) - \mathbf{R}_a^T \mathbf{R}_b (\delta\vartheta \mathbf{w}_\vartheta + \delta\varphi \mathbf{w}_\varphi) \quad (9.155)$$

with

$$\begin{aligned} \mathbf{w}_\vartheta &= \left(\mathbf{I} + \exp(\vartheta \mathbf{e}_2 \times) \right)^T \exp(\varphi \mathbf{e}_1 \times)^T \mathbf{e}_2 \\ &= \sin \varphi \sin \vartheta \mathbf{e}_1 + (1 + \cos \varphi) \mathbf{e}_2 - \sin \varphi \cos \vartheta \mathbf{e}_3 \end{aligned} \quad (9.156a)$$

$$\begin{aligned} \mathbf{w}_\varphi &= \exp(\vartheta \mathbf{e}_2 \times)^T \mathbf{e}_1 \\ &= \cos \vartheta \mathbf{e}_1 + \sin \vartheta \mathbf{e}_3. \end{aligned} \quad (9.156b)$$

Linearization

$$\begin{aligned} &\begin{bmatrix} -(\mathbf{R}_a \hat{\boldsymbol{\lambda}}) \times & \mathbf{0} & \mathbf{R}_a \\ (\mathbf{R}_a \hat{\boldsymbol{\lambda}}) \times & \mathbf{0} & -\mathbf{R}_a \\ \mathbf{R}_a^T & -\mathbf{R}_a^T & \mathbf{0} \\ -((\mathbf{R}_b \mathbf{w}_\vartheta) \times (\mathbf{R}_a \hat{\boldsymbol{\lambda}}))^T & ((\mathbf{R}_b \mathbf{w}_\vartheta) \times (\mathbf{R}_a \hat{\boldsymbol{\lambda}}))^T & (\mathbf{R}_a^T \mathbf{R}_b \mathbf{w}_\vartheta)^T \\ -((\mathbf{R}_b \mathbf{w}_\varphi) \times (\mathbf{R}_a \hat{\boldsymbol{\lambda}}))^T & ((\mathbf{R}_b \mathbf{w}_\varphi) \times (\mathbf{R}_a \hat{\boldsymbol{\lambda}}))^T & (\mathbf{R}_a^T \mathbf{R}_b \mathbf{w}_\varphi)^T \end{bmatrix} \begin{Bmatrix} \theta_{a\Delta} \\ \theta_{b\Delta} \\ \Delta \hat{\boldsymbol{\lambda}} \end{Bmatrix} \\ &\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{R}_a^T \mathbf{R}_b \mathbf{w}_\vartheta & \mathbf{R}_a^T \mathbf{R}_b \mathbf{w}_\varphi \\ (\mathbf{R}_a^T \mathbf{R}_b \mathbf{w}_{\vartheta/\vartheta})^T \hat{\boldsymbol{\lambda}} & (\mathbf{R}_a^T \mathbf{R}_b \mathbf{w}_{\vartheta/\varphi})^T \hat{\boldsymbol{\lambda}} \\ (\mathbf{R}_a^T \mathbf{R}_b \mathbf{w}_{\varphi/\vartheta})^T \hat{\boldsymbol{\lambda}} & (\mathbf{R}_a^T \mathbf{R}_b \mathbf{w}_{\varphi/\varphi})^T \hat{\boldsymbol{\lambda}} \end{bmatrix} \begin{Bmatrix} \Delta \vartheta \\ \Delta \varphi \end{Bmatrix} \\ &= \begin{Bmatrix} -\mathbf{R}_a \hat{\boldsymbol{\lambda}} \\ \mathbf{R}_a \hat{\boldsymbol{\lambda}} \\ \boldsymbol{\theta} \\ -(\mathbf{R}_a^T \mathbf{R}_b \mathbf{w}_\vartheta)^T \hat{\boldsymbol{\lambda}} \\ -(\mathbf{R}_a^T \mathbf{R}_b \mathbf{w}_\varphi)^T \hat{\boldsymbol{\lambda}} \end{Bmatrix} \end{aligned} \quad (9.157)$$

where

$$\mathbf{w}_{\vartheta/\vartheta} = \sin \varphi \cos \vartheta \mathbf{e}_1 + \sin \varphi \sin \vartheta \mathbf{e}_3 \quad (9.158a)$$

$$\mathbf{w}_{\vartheta/\varphi} = \cos \varphi \sin \vartheta \mathbf{e}_1 - \sin \varphi \mathbf{e}_2 - \cos \varphi \cos \vartheta \mathbf{e}_3 \quad (9.158b)$$

$$\mathbf{w}_{\varphi/\vartheta} = -\sin \vartheta \mathbf{e}_1 + \cos \vartheta \mathbf{e}_3 \quad (9.158c)$$

$$\mathbf{w}_{\varphi/\varphi} = \mathbf{0}. \quad (9.158d)$$

9.1.15 Screw Joint

Author: Mauro Manetti and Marco Morandini

The screw joint is a combination of:

- an in line joint, that constrains one point to move along a line attached to another point:

$$\mathbf{e}_{1hx}^T (\mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1 - \mathbf{f}_1) = 0$$

$$\mathbf{e}_{1hy}^T (\mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1 - \mathbf{f}_1) = 0$$

where \mathbf{e}_{1hi} is the i -th column of matrix $\mathbf{R}_1 \tilde{\mathbf{R}}_{1h}$

$$\mathbf{e}_{jhi} = (\mathbf{R}_j \tilde{\mathbf{R}}_{jh}) [i] \quad (9.159)$$

\mathbf{x}_j is the position of node j and \mathbf{f}_j is the offset of the joint with respect node j , both in the absolute coordinate system; the offset is constant with respect to the reference frame of the joint, so $\mathbf{f}_j = \mathbf{R}_j \tilde{\mathbf{f}}_j$.

- a revolute rotation joint, that constrains the relative orientation of two nodes to be a rotation about an axis that is fixed with respect to the two bodies:

$$\begin{aligned}\mathbf{e}_{2hx}^T \mathbf{e}_{1hz} &= 0 \\ \mathbf{e}_{2hy}^T \mathbf{e}_{1hz} &= 0\end{aligned}$$

- a linear relationship between the relative rotation about the common axis and the relative position along the common axis:

$$\frac{p}{2\pi}(\theta - \theta_0) - \mathbf{e}_{1hz}^T (\mathbf{d} - \mathbf{d}_0) = 0 \quad (9.160)$$

where the nodes distance vectors \mathbf{d} and \mathbf{d}_0 can be written as:

$$\begin{aligned}\mathbf{d} &= \mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1 - \mathbf{f}_1 \\ \mathbf{d}_0 &= \mathbf{x}_{20} + \mathbf{f}_{20} - \mathbf{x}_{10} - \mathbf{f}_{10}\end{aligned}$$

while p represents the distance between the nodes along axis \mathbf{e}_{1hz} corresponding to one revolution about the same axis.

The rotation θ is formulated as

$$\begin{aligned}\theta &= \tilde{\mathbf{R}}_{1h3}^T \text{ax}(\exp^{-1}(\mathbf{R}_1^T \mathbf{R}_2)) \\ &= \tilde{\mathbf{R}}_{1h3}^T \mathbf{R}_1^T \text{ax}(\exp^{-1}(\mathbf{R}_2 \mathbf{R}_1^T)) \\ &= \mathbf{e}_{1hz}^T \text{ax}(\exp^{-1}(\mathbf{R}_2 \mathbf{R}_1^T))\end{aligned} \quad (9.161)$$

where $\tilde{\mathbf{R}}_{1h3}$ is the third axis of the constant relative orientation of node 1, while \mathbf{e}_{1hz} is the screw joint axis. To overcome the limitation $|\theta| < \pi$, an appropriate unwrap angle function has been implemented in the code, so allowing to compute the right residual of Equation (9.160).

The virtual perturbation of Equation (9.161) can be expressed as

$$\delta\theta = \tilde{\mathbf{R}}_{1h3}^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_1^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) \quad (9.162)$$

so the virtual perturbation of Equation (9.160)

$$\begin{aligned}\frac{p}{2\pi} \mathbf{e}_{1hz}^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_1^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) - \mathbf{e}_{1hz}^T \mathbf{d} \times \boldsymbol{\theta}_{1\delta} \\ - \mathbf{e}_{1hz}^T (\delta \mathbf{x}_2 - \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} - \delta \mathbf{x}_1 + \mathbf{f}_1 \times \boldsymbol{\theta}_{1\delta}) = 0\end{aligned} \quad (9.163)$$

yields the contribution of the constraint to the forces and moments acting on the constrained nodes:

$$\mathbf{F}_1 = \mathbf{e}_{1hz} \lambda \quad (9.164)$$

$$\mathbf{C}_1 = \left(-\frac{p}{2\pi} \mathbf{R}_1 \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \tilde{\mathbf{R}}_{1h3} + \mathbf{f}_1 \times \mathbf{e}_{1hz} \right) \lambda \quad (9.165)$$

$$\mathbf{F}_2 = -\mathbf{e}_{1hz} \lambda \quad (9.166)$$

$$\mathbf{C}_2 = \left(\frac{p}{2\pi} \mathbf{R}_1 \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \tilde{\mathbf{R}}_{1h3} - \mathbf{f}_2 \times \mathbf{e}_{1hz} \right) \lambda \quad (9.167)$$

where λ is the Lagrange multiplier that here assumes the meaning of reaction force along the screw axis.

The problem linearization requires the already computed virtual perturbation of the constraint equation (see Equation (9.163)) and the constraint forces and couples virtual perturbation:

$$\delta \mathbf{F}_1 = -\lambda \mathbf{e}_{1hz} \times \boldsymbol{\theta}_{1\delta} + \mathbf{e}_{1hz} \delta \lambda \quad (9.168)$$

$$\begin{aligned} \delta \mathbf{C}_1 = & \lambda \left[\frac{p}{2\pi} \left(\mathbf{R}_1 \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \tilde{\mathbf{R}}_{1h3} \right) \times \boldsymbol{\theta}_{1\delta} - \right. \\ & \left. \frac{p}{2\pi} \mathbf{R}_1 \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \mathbf{L} \left(-\boldsymbol{\theta}, \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \tilde{\mathbf{R}}_{1h3} \right) \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_1^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) + \right. \\ & \left. \mathbf{e}_{1hz} \times \mathbf{f}_1 \times \boldsymbol{\theta}_{1\delta} - \mathbf{f}_1 \times \mathbf{e}_{1hz} \times \boldsymbol{\theta}_{1\delta} \right] + \\ & \frac{C_1}{\lambda} \delta \lambda \end{aligned} \quad (9.169)$$

$$\delta \mathbf{F}_2 = \lambda \mathbf{e}_{1hz} \times \boldsymbol{\theta}_{1\delta} - \mathbf{e}_{1hz} \delta \lambda \quad (9.170)$$

$$\begin{aligned} \delta \mathbf{C}_2 = & \lambda \left[-\frac{p}{2\pi} \left(\mathbf{R}_1 \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \tilde{\mathbf{R}}_{1h3} \right) \times \boldsymbol{\theta}_{1\delta} + \right. \\ & \left. \frac{p}{2\pi} \mathbf{R}_1 \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \mathbf{L} \left(-\boldsymbol{\theta}, \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \tilde{\mathbf{R}}_{1h3} \right) \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_1^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) - \right. \\ & \left. \mathbf{e}_{1hz} \times \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} + \mathbf{f}_2 \times \mathbf{e}_{1hz} \times \boldsymbol{\theta}_{1\delta} \right] + \\ & \frac{C_2}{\lambda} \delta \lambda \end{aligned} \quad (9.171)$$

where the operator $\mathbf{L}()$ has been introduced according to the following definition

$$\delta \boldsymbol{\Gamma}(\boldsymbol{\theta})^T \mathbf{a} = -\mathbf{L}(-\boldsymbol{\theta}, \mathbf{a}) \delta \boldsymbol{\theta} \quad (9.172)$$

which can be manipulated to obtain the needed relation

$$\begin{aligned} \delta (\boldsymbol{\Gamma}(\boldsymbol{\theta})^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \mathbf{a}) = & \delta \boldsymbol{\Gamma}(\boldsymbol{\theta})^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \mathbf{a} + \\ & \boldsymbol{\Gamma}(\boldsymbol{\theta})^T \delta \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \mathbf{a} + \\ & \boldsymbol{\Gamma}(\boldsymbol{\theta})^T \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \delta \mathbf{a} = \delta \mathbf{a} \end{aligned} \quad (9.173)$$

$$\delta \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \mathbf{a} = \delta \boldsymbol{\Gamma}(\boldsymbol{\theta})^T \mathbf{L}(-\boldsymbol{\theta}, \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \mathbf{a}) \delta \boldsymbol{\theta} \quad (9.174)$$

Now a contribution useful to obtain previous results will be reported

$$\begin{aligned} \delta \left(\mathbf{R}_1 \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \tilde{\mathbf{R}}_{1h3} \right) = & \delta \mathbf{R}_1 \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \tilde{\mathbf{R}}_{1h3} + \mathbf{R}_1 \delta \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \tilde{\mathbf{R}}_{1h3} \\ = & - \left(\mathbf{R}_1 \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \tilde{\mathbf{R}}_{1h3} \right)_x \boldsymbol{\theta}_{1\delta} + \mathbf{R}_1 \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \mathbf{L}(-\boldsymbol{\theta}, \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-T} \tilde{\mathbf{R}}_{1h3}) \delta \boldsymbol{\theta} \end{aligned}$$

This joint formulation can be improved to take into account the presence of friction. Figures (9.1) and (9.2) show sketches useful to identify the forces acting on the screw thread. In the figures the case of a screw which is raising a load \mathbf{F} is presented, the equilibrium equation can be written for the vertical and the horizontal directions to obtain

$$\mathbf{F} + \mu \mathbf{F}_n \sin \alpha = \mathbf{F}_n \cos \gamma_n \cos \alpha \quad (9.175)$$

$$\mathbf{P} = \mu \mathbf{F}_n \cos \alpha + \mathbf{F}_n \cos \gamma_n \sin \alpha \quad (9.176)$$

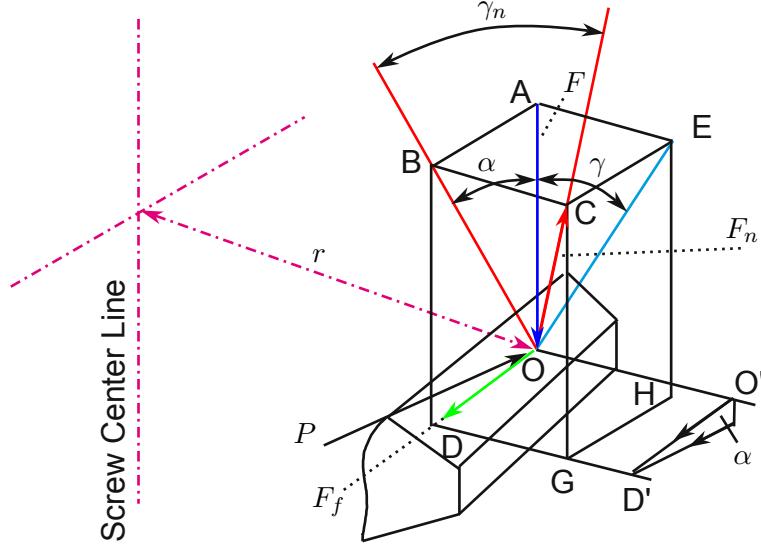


Figure 9.1: 3D screw thread sketch.

where \mathbf{F} is the load, or the force, along the screw axis, \mathbf{P} the force acting to rotate the screw, i.e. the torque moment over the mean thread radius r , \mathbf{F}_n is the thread normal reaction, α the lead angle, and γ_n the angle between the vectors \mathbf{OC} and \mathbf{OB} represented in Figure (9.1). Inside the code all the screw friction formulation is developed in function of the angle γ_n for convenience, anyway the usual screw informations relate the thread angle 2γ , so the joint requires as input the half thread angle γ . Looking at the Figure 9.1 it is possible to find the relation between these two angles

$$BC = AE = OA \tan \gamma = OB \cos \alpha \tan \gamma \quad (9.177)$$

so

$$\tan \gamma_n = \frac{BC}{OB} = \cos \alpha \tan \gamma \quad \Rightarrow \quad \gamma_n = \tan^{-1}(\cos \alpha \tan \gamma). \quad (9.178)$$

From Equations (9.175) and (9.176) is possible to retrieve the relation between the screw raising torque C_r and the axial force \mathbf{F}

$$C_r = r\mathbf{F} \left(\frac{\cos \gamma_n \sin \alpha + \mu \cos \alpha}{\cos \gamma_n \cos \alpha - \mu \sin \alpha} \right). \quad (9.179)$$

It is easy to find the previous relation in presence of a lowering torque

$$C_l = r\mathbf{F} \left(\frac{\cos \gamma_n \sin \alpha - \mu \cos \alpha}{\cos \gamma_n \cos \alpha + \mu \sin \alpha} \right). \quad (9.180)$$

from which it is possible to understand the torque dependency from the versus of the friction force $\mu \mathbf{F}_n$. A general discriminant to apply the right formula can be linked to the versus of the relative velocity v between the internal and external thread and the sign of the constraint Lagrange multiplier λ which

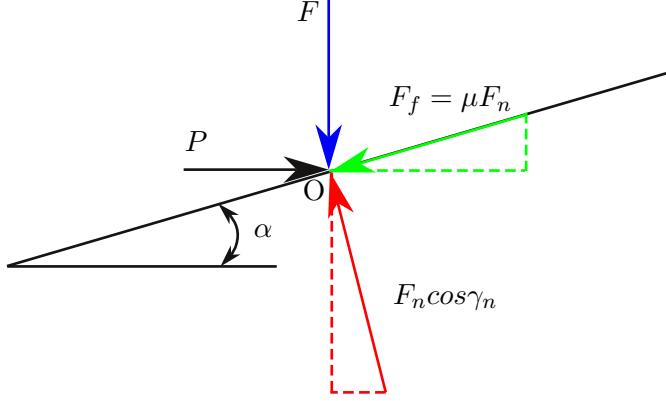


Figure 9.2: 2D screw thread sketch

represents the versus of the axial screw force

$$\mathbf{C} = r\mathbf{F} \left(\frac{\cos \gamma_n \sin \alpha + \text{sign}(v\lambda) \mu \cos \alpha}{\cos \gamma_n \cos \alpha - \text{sign}(v\lambda) \mu \sin \alpha} \right). \quad (9.181)$$

Inside the code $\text{sign}(v\lambda)$ will be embedded in the friction coefficient computation $\mu = \mu(v, \mathbf{F})$, so in the following for brevity it will be removed from equations introducing the notation $\mathbf{C} = \mathbf{C}(\mu)$.

The torque acting on the screw for the friction presence can now be computed as

$$\mathbf{C}_{frc}(\mu) = \mathbf{C}(\mu) - \mathbf{C}(0) = r\mathbf{F} \left[\frac{\mu \sec \gamma_n (1 + \tan^2 \alpha)}{1 - \mu \sec \gamma_n \tan \alpha} \right]. \quad (9.182)$$

Equation (9.182) can be used to add the torque friction contribution to the couples acting on the constrained nodes reported in Equations (9.165) and (9.167)

$$\mathbf{C}_{1_{frc}} = \mathbf{C}_1 - \mathbf{C}_{frc}(\mu) = \mathbf{C}_1 - r\mathbf{F}_1 \left[\frac{\mu \sec \gamma_n (1 + \tan^2 \alpha)}{1 - \mu \sec \gamma_n \tan \alpha} \right] \quad (9.183)$$

$$\mathbf{C}_{2_{frc}} = \mathbf{C}_2 - \mathbf{C}_{frc}(\mu) = \mathbf{C}_2 - r\mathbf{F}_2 \left[\frac{\mu \sec \gamma_n (1 + \tan^2 \alpha)}{1 - \mu \sec \gamma_n \tan \alpha} \right] \quad (9.184)$$

At this point the only further required informations regard the relative velocity v and the virtual variation of the friction torque contribution. Starting from the already used formula

$$\delta\boldsymbol{\theta} = \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_1^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) \quad (9.185)$$

it is possible to write the following relation

$$\dot{\boldsymbol{\theta}} = \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_1^T (\boldsymbol{\Gamma}(\boldsymbol{\theta}_2) \dot{\boldsymbol{\theta}}_2 - \boldsymbol{\Gamma}(\boldsymbol{\theta}_1) \dot{\boldsymbol{\theta}}_1). \quad (9.186)$$

The angular velocity vector is defined as

$$\boldsymbol{\omega}_\theta = \boldsymbol{\Gamma}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}} = \boldsymbol{\Gamma}(\boldsymbol{\theta}) \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_1^T (\boldsymbol{\Gamma}(\boldsymbol{\theta}_2) \dot{\boldsymbol{\theta}}_2 - \boldsymbol{\Gamma}(\boldsymbol{\theta}_1) \dot{\boldsymbol{\theta}}_1) = \mathbf{R}_1^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \quad (9.187)$$

from which the angular velocity modulus with sign can be obtained

$$\omega_\theta = \tilde{\mathbf{R}}_{1h3}^T \boldsymbol{\omega}_\theta = \tilde{\mathbf{R}}_{1h3}^T \mathbf{R}_1^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1). \quad (9.188)$$

Now the relative velocity on the screw thread, along the friction force direction, can be written as

$$v = \omega_\theta r \cos \alpha = \tilde{\mathbf{R}}_{1h3}^T \mathbf{R}_1^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) r \cos \alpha. \quad (9.189)$$

The virtual perturbation of the friction couple is required to write the linearized problem

$$\begin{aligned} \delta \mathbf{C}_{frc} &= \delta \mathbf{F} r \left[\frac{\mu \sec \gamma_n (1 + \tan^2 \alpha)}{1 - \mu \sec \gamma_n \tan \alpha} \right] + \\ &\quad \mathbf{F} r \left[\frac{\sec \gamma_n (1 + \tan^2 \alpha)}{1 - \mu \sec \gamma_n \tan \alpha} + \frac{\mu \sec \gamma_n (1 + \tan^2 \alpha)}{(1 - \mu \sec \gamma_n \tan \alpha)^2} \sec \gamma_n \tan \alpha \right] \delta \mu \\ &= r \left[\frac{\mu \sec \gamma_n (1 + \tan^2 \alpha)}{1 - \mu \sec \gamma_n \tan \alpha} \right] \delta \mathbf{F} + \mathbf{F} r \left[\frac{\mu \sec \gamma_n (1 + \tan^2 \alpha)}{(1 - \mu \sec \gamma_n \tan \alpha)^2} \right] \delta \mu \end{aligned} \quad (9.190)$$

where the friction coefficient virtual variation is automatically computed by the code, once provided the F and μ virtual perturbations

$$\delta \mu = \delta \mu(\delta \mathbf{F}, \delta v). \quad (9.191)$$

Anyway the screw axial forces virtual variations have already been reported in Equations 9.168 and 9.170, so to complete the problem description is just required the definition of the relative velocity virtual perturbation

$$\begin{aligned} \delta v &= \tilde{\mathbf{R}}_{1h3}^T \delta \mathbf{R}_1^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) r \cos \alpha + \tilde{\mathbf{R}}_{1h3}^T \mathbf{R}_1^T (\delta \boldsymbol{\omega}_2 - \delta \boldsymbol{\omega}_1) r \cos \alpha \\ &= r \cos \alpha \tilde{\mathbf{R}}_{1h3}^T \mathbf{R}_1^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \boldsymbol{\theta}_{1\delta} + r \cos \alpha \tilde{\mathbf{R}}_{1h3}^T (\delta \boldsymbol{\omega}_2 - \delta \boldsymbol{\omega}_1). \end{aligned} \quad (9.192)$$

Physics. In other words, node 1 is the screw and node 2 is the bolt; neglecting the offset, the force along the screw axis is related to the couple about the same axis by the relationship

$$C = -\frac{p}{2\pi} F \quad (9.193)$$

which results from a power balance

$$C \omega_\theta + F v_{lin} = 0 \quad (9.194)$$

in terms of relative linear (v_{lin}) and angular (ω_θ) velocity, with the kinematic relationship

$$v_{lin} = \frac{p}{2\pi} \omega_\theta \quad (9.195)$$

9.1.16 Strapdown Sensor

This constraint imposes the acceleration and the angular velocity of a node in a relative frame attached to the node itself.

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}_n + \dot{\boldsymbol{\omega}}_n \times \mathbf{b} + \boldsymbol{\omega}_n \times \boldsymbol{\omega}_n \times \mathbf{b} \quad (9.196a)$$

$$\boldsymbol{\omega} = \boldsymbol{\omega}_n \quad (9.196b)$$

where $\mathbf{b} = \mathbf{R}_n \tilde{\mathbf{b}}$ is the offset of the accelerometer from the node.

The acceleration and the angular velocity in the reference frame of the node, as output from the strapdown sensor, respectively are $\bar{\mathbf{x}}$ and $\bar{\boldsymbol{\omega}}$. The constraint equations are

$$\ddot{\mathbf{x}} = \mathbf{R}_n \bar{\mathbf{x}} \quad (9.197a)$$

$$\boldsymbol{\omega} = \mathbf{R}_n \bar{\boldsymbol{\omega}} \quad (9.197b)$$

The acceleration of the point where the strapdown sensor is placed is

$$\begin{aligned} \ddot{\mathbf{x}} &= \left(\frac{1}{m} \mathbf{I} + \mathbf{f} \times {}^T \mathbf{J}_{\text{CM}}^{-1} \mathbf{f} \times \right) \dot{\boldsymbol{\beta}} - \mathbf{f} \times {}^T \mathbf{J}_{\text{CM}}^{-1} \dot{\boldsymbol{\gamma}} - \mathbf{f} \times {}^T \mathbf{J}_{\text{CM}}^{-1} (\dot{\mathbf{x}}_n \times \boldsymbol{\beta} - \boldsymbol{\omega}_n \times \mathbf{J}_{\text{CM}} \boldsymbol{\omega}_n) \\ &\quad + \mathbf{b} \times {}^T \mathbf{J}_{\text{CM}}^{-1} \left(\dot{\boldsymbol{\gamma}} - \mathbf{f} \times \dot{\boldsymbol{\beta}} + \dot{\mathbf{x}}_n \times \boldsymbol{\beta} - \boldsymbol{\omega}_n \times \mathbf{J}_{\text{CM}} \boldsymbol{\omega}_n \right) + \boldsymbol{\omega}_n \times \boldsymbol{\omega}_n \times \mathbf{b} \\ &= \left(\frac{1}{m} \mathbf{I} + (\mathbf{f} - \mathbf{b}) \times {}^T \mathbf{J}_{\text{CM}}^{-1} \mathbf{f} \times \right) \dot{\boldsymbol{\beta}} - (\mathbf{f} - \mathbf{b}) \times {}^T \mathbf{J}_{\text{CM}}^{-1} \dot{\boldsymbol{\gamma}} \\ &\quad - (\mathbf{f} - \mathbf{b}) \times {}^T \mathbf{J}_{\text{CM}}^{-1} (\dot{\mathbf{x}}_n \times \boldsymbol{\beta} - \boldsymbol{\omega}_n \times \mathbf{J}_{\text{CM}} \boldsymbol{\omega}_n) + \boldsymbol{\omega}_n \times \boldsymbol{\omega}_n \times \mathbf{b} \end{aligned} \quad (9.198)$$

Note: this constraint is extremely simpler when the node is in the center of mass ($\mathbf{f} = \mathbf{0}$), and the strapdown sensor is in the node ($\mathbf{b} = \mathbf{0}$).

9.2 Deformable Constraints

Definitions

$$\begin{aligned} \mathbf{f}_1 &= \mathbf{R}_1 \tilde{\mathbf{f}}_1 \\ \mathbf{f}_2 &= \mathbf{R}_2 \tilde{\mathbf{f}}_2 \\ \mathbf{R}_{1h} &= \mathbf{R}_1 \tilde{\mathbf{R}}_{1h} \\ \mathbf{R}_{2h} &= \mathbf{R}_2 \tilde{\mathbf{R}}_{2h} \end{aligned}$$

where \mathbf{R}_i is the current orientation of node i , $\tilde{\mathbf{R}}_{ih}$ is a constant re-orientation of the joint with respect to node i , so \mathbf{R}_{ih} is the orientation of the joint with respect to the global frame; $\tilde{\mathbf{f}}_i$ is the offset of the joint with respect to node i in the node reference frame, so \mathbf{f}_i is the offset of the joint with respect to node i in the global frame, and $\mathbf{x}_i + \mathbf{f}_i$ is the position of the joint with respect to the global frame.

Equilibrium equations are obtained by means of the Virtual Work Principle (VWP). While the unknown orientation parameters depend on the parametrization in use, and typically are Gibbs-Rodrigues parameters according to the updated-updated approach, equilibrium is written in terms of the equations conjugated to perturbations of relative rotation, namely:

$$\delta \mathcal{L} = \sum \delta \mathbf{x}_i^T \mathbf{F}_i + \theta_{i\delta} \mathbf{M}_i = 0 \quad (9.199)$$

9.2.1 Rod With Offsets

Distance vector between pin points

$$\mathbf{l} = \mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1 - \mathbf{f}_1 \quad (9.200)$$

scalar distance

$$l = \sqrt{\mathbf{l}^T \mathbf{l}} \quad (9.201)$$

strain

$$\varepsilon = \frac{l}{l_0} - 1 \quad (9.202)$$

strain rate

$$\dot{\varepsilon} = \frac{\dot{l}}{l_0} \quad (9.203)$$

where

$$\dot{l} = \frac{1}{l} \mathbf{l}^T \dot{\mathbf{l}} \quad (9.204a)$$

$$\dot{\mathbf{l}} = \dot{\mathbf{x}}_2 + \boldsymbol{\omega}_2 \times \mathbf{f}_2 - \dot{\mathbf{x}}_1 - \boldsymbol{\omega}_1 \times \mathbf{f}_1 \quad (9.204b)$$

scalar force

$$f = f(\varepsilon, \dot{\varepsilon}) \quad (9.205)$$

nodal forces and moments result from the VWP according to

$$\delta \mathcal{L} = \delta l^T f \quad (9.206a)$$

$$= \delta l^T \frac{\mathbf{l}}{l} f \quad (9.206b)$$

$$= (\delta \mathbf{x}_2 - \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} - \delta \mathbf{x}_1 + \mathbf{f}_1 \times \boldsymbol{\theta}_{1\delta})^T \frac{\mathbf{l}}{l} f \quad (9.206c)$$

force vector

$$\mathbf{F} = \frac{\mathbf{l}}{l} f \quad (9.207)$$

nodal forces and moments

$$\mathbf{F}_1 = -\mathbf{F} \quad (9.208a)$$

$$\mathbf{M}_1 = -\mathbf{f}_1 \times \mathbf{F} \quad (9.208b)$$

$$\mathbf{F}_2 = \mathbf{F} \quad (9.208c)$$

$$\mathbf{M}_2 = \mathbf{f}_2 \times \mathbf{F} \quad (9.208d)$$

equation linearization

$$\delta \mathbf{F}_1 = -\delta \mathbf{F} \quad (9.209a)$$

$$\delta \mathbf{M}_1 = -\mathbf{f}_1 \times \delta \mathbf{F} - \mathbf{F} \times \mathbf{f}_1 \times \boldsymbol{\theta}_{1\delta} \quad (9.209b)$$

$$\delta \mathbf{F}_2 = \delta \mathbf{F} \quad (9.209c)$$

$$\delta \mathbf{M}_2 = \mathbf{f}_2 \times \delta \mathbf{F} + \mathbf{F} \times \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} \quad (9.209d)$$

force linearization

$$\delta \mathbf{F} = \frac{f}{l} \left(\mathbf{I} - \frac{\mathbf{l} \mathbf{l}^T}{l^2} \right) \delta l + \frac{\mathbf{l}}{l} \delta f \quad (9.210)$$

scalar force linearization

$$\delta f = \frac{\partial f}{\partial \varepsilon} \delta \varepsilon + \frac{\partial f}{\partial \dot{\varepsilon}} \delta \dot{\varepsilon} \quad (9.211)$$

strain linearization

$$\delta\varepsilon = \frac{1}{l_0} \delta l \quad (9.212)$$

where

$$\delta l = \frac{1}{l} \mathbf{l}^T \delta \mathbf{l} \quad (9.213a)$$

$$\delta \mathbf{l} = \delta \dot{\mathbf{x}}_2 - \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} - \delta \dot{\mathbf{x}}_1 + \mathbf{f}_1 \times \boldsymbol{\theta}_{1\delta} \quad (9.213b)$$

strain rate linearization

$$\delta\dot{\varepsilon} = \frac{1}{l_0} \delta \dot{l} \quad (9.214)$$

where

$$\delta \dot{l} = \frac{\mathbf{l}^T}{l} \delta \dot{\mathbf{l}} + \frac{\dot{\mathbf{l}}^T}{l} \left(\mathbf{I} - \frac{\mathbf{l}\mathbf{l}^T}{l^2} \right) \delta \mathbf{l} \quad (9.215a)$$

$$\begin{aligned} \delta \dot{\mathbf{l}} = & \delta \dot{\mathbf{x}}_2 - \mathbf{f}_2 \times \delta \boldsymbol{\omega}_2 - \boldsymbol{\omega}_2 \times \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} \\ & - \delta \dot{\mathbf{x}}_1 + \mathbf{f}_1 \times \delta \boldsymbol{\omega}_1 + \boldsymbol{\omega}_1 \times \mathbf{f}_1 \times \boldsymbol{\theta}_{1\delta} \end{aligned} \quad (9.215b)$$

to summarize:

$$\delta \mathbf{F} = \mathbf{K}_l \delta \mathbf{l} + \mathbf{K}_i \delta \dot{\mathbf{l}} \quad (9.216)$$

with

$$\mathbf{K}_l = \frac{f}{l} \mathbf{I} + \left(\frac{1}{l^2 l_0} \frac{\partial f}{\partial \varepsilon} - \frac{\dot{\varepsilon}}{l^3} \frac{\partial f}{\partial \dot{\varepsilon}} - \frac{f}{l^3} \right) \mathbf{U}^T + \frac{1}{l^2 l_0} \frac{\partial f}{\partial \dot{\varepsilon}} \dot{\mathbf{U}}^T \quad (9.217a)$$

$$\mathbf{K}_i = \frac{1}{l^2 l_0} \frac{\partial f}{\partial \dot{\varepsilon}} \mathbf{U}^T \quad (9.217b)$$

according to the simplifications of the updated-updated approach,

$$\boldsymbol{\theta}_{i\delta} \stackrel{\text{uu}}{=} \delta \mathbf{g}_i \quad (9.218a)$$

$$\delta \boldsymbol{\omega}_i \stackrel{\text{uu}}{=} \delta \dot{\mathbf{g}}_i - \boldsymbol{\omega} \times \delta \mathbf{g}_i \quad (9.218b)$$

recalling that

$$\delta z = c \delta \dot{z} \quad (9.219)$$

the linearization of the force becomes

$$\begin{aligned} \delta \mathbf{F} = & (c \mathbf{K}_l + \mathbf{K}_i) \delta \dot{\mathbf{x}}_2 \\ & + (- (c \mathbf{K}_l + \mathbf{K}_i) \mathbf{f}_2 \times + c \mathbf{K}_i (\mathbf{f}_2 \times \boldsymbol{\omega}_2) \times) \delta \dot{\mathbf{g}}_2 \\ & - (c \mathbf{K}_l + \mathbf{K}_i) \delta \dot{\mathbf{x}}_1 \\ & - (- (c \mathbf{K}_l + \mathbf{K}_i) \mathbf{f}_1 \times + c \mathbf{K}_i (\mathbf{f}_1 \times \boldsymbol{\omega}_1) \times) \delta \dot{\mathbf{g}}_1 \end{aligned} \quad (9.220)$$

9.2.2 Deformable Hinge

The deformable hinge applies to two nodes an internal moment that may depend on their relative orientation and angular velocity by means of a 3D constitutive law. It is discussed in [6]. Two variants of this joint are presented:

- the one historically implemented in MBDyn, called “attached”, considers the constitutive law and the resulting moment attached to node 1;
- the other one, called “invariant”, defines an intermediate orientation that is halfway between that of the two nodes, and considers the constitutive law and the resulting moment attached to that intermediate orientation; as a consequence, the resulting internal moment does not depend on the node sequence even when anisotropic constitutive laws are considered.

The relative rotation vector is computed in analogy with the `drive hinge` joint (`DriveHingeJoint`).

$$\boldsymbol{\theta} = \text{ax}(\exp^{-1}(\mathbf{R}_{1h}^T \mathbf{R}_{2h})) \quad (9.221)$$

where \mathbf{R}_{1h} , \mathbf{R}_{2h} are the matrices that express the orientation of each side of the hinge in the global reference frame, defined as

$$\mathbf{R}_{1h} = \mathbf{R}_1 \tilde{\mathbf{R}}_{1h} \quad (9.222a)$$

$$\mathbf{R}_{2h} = \mathbf{R}_2 \tilde{\mathbf{R}}_{2h} \quad (9.222b)$$

and $\tilde{\mathbf{R}}_{1h}$, $\tilde{\mathbf{R}}_{2h}$ are the matrices that express the orientation of each side of the hinge with respect to the corresponding node. The perturbation of \mathbf{R}_{1h} , \mathbf{R}_{2h} yields

$$\delta \mathbf{R}_{1h} = \boldsymbol{\theta}_{1\delta} \times \mathbf{R}_1 \tilde{\mathbf{R}}_{1h} = \boldsymbol{\theta}_{1\delta} \times \mathbf{R}_{1h} \quad (9.223a)$$

$$\delta \mathbf{R}_{2h} = \boldsymbol{\theta}_{2\delta} \times \mathbf{R}_2 \tilde{\mathbf{R}}_{2h} = \boldsymbol{\theta}_{2\delta} \times \mathbf{R}_{2h} \quad (9.223b)$$

since matrices $\tilde{\mathbf{R}}_{1h}$, $\tilde{\mathbf{R}}_{2h}$ are constant.

Since the relative orientation $\mathbf{R}_{1h}^T \mathbf{R}_{2h}$ is a rotation about an axis parallel to $\boldsymbol{\theta}$, $\boldsymbol{\theta}$ itself does not change when it is referred to any intermediate orientation between \mathbf{I} and $\mathbf{R}_{1h}^T \mathbf{R}_{2h}$, namely

$$\mathbf{R}_{1h}^T \mathbf{R}_{2h} \boldsymbol{\theta} = \boldsymbol{\theta}. \quad (9.224)$$

The same applies for any relative orientation tensor built from a rotation vector $\alpha \boldsymbol{\theta}$ parallel to $\boldsymbol{\theta}$, whatever value the scalar α assumes.

The relative angular velocity is defined as the derivative of the relative orientation matrix

$$\begin{aligned} \boldsymbol{\omega} \times &= \frac{d}{dt} (\mathbf{R}_{1h}^T \mathbf{R}_{2h}) (\mathbf{R}_{1h}^T \mathbf{R}_{2h})^T \\ &= \left(\dot{\mathbf{R}}_{1h}^T \mathbf{R}_{2h} + \mathbf{R}_{1h}^T \dot{\mathbf{R}}_{2h} \right) \mathbf{R}_{2h}^T \mathbf{R}_{1h} \\ &= \mathbf{R}_{1h}^T \boldsymbol{\omega}_1 \times^T \mathbf{R}_{1h} + \mathbf{R}_{1h}^T \boldsymbol{\omega}_2 \times \mathbf{R}_{1h} \\ &= \mathbf{R}_{1h}^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \mathbf{R}_{1h} \end{aligned} \quad (9.225)$$

so

$$\boldsymbol{\omega} = \mathbf{R}_{1h}^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \quad (9.226)$$

Attached Deformable Hinge

The constitutive law is defined attached to the reference frame of node 1; the value of the relative rotation vector, $\boldsymbol{\theta}$, does not vary, according to what stated earlier.

The perturbation of $\boldsymbol{\theta}$ yields

$$\boldsymbol{\theta}_\delta = \mathbf{R}_{1h}^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) \quad (9.227)$$

which, according to the simplifications of the updated-updated approach, becomes

$$\boldsymbol{\theta}_\delta \stackrel{\text{uu}}{=} \mathbf{R}_{1h}^T (\delta \mathbf{g}_2 - \delta \mathbf{g}_1) \quad (9.228)$$

The perturbation of the relative angular velocity of Equation (9.226), according to the simplifications of the updated-updated approach, yields

$$\begin{aligned} \delta \boldsymbol{\omega} &= \mathbf{R}_{1h}^T ((\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \boldsymbol{\theta}_{1\delta} + \delta \boldsymbol{\omega}_2 - \delta \boldsymbol{\omega}_1) \\ &\stackrel{\text{uu}}{=} \mathbf{R}_{1h}^T ((\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \delta \mathbf{g}_1 + \delta \dot{\mathbf{g}}_2 - \boldsymbol{\omega}_2 \times \delta \mathbf{g}_2 - \delta \dot{\mathbf{g}}_1 + \boldsymbol{\omega}_1 \times \delta \mathbf{g}_1) \\ &= \mathbf{R}_{1h}^T (\delta \dot{\mathbf{g}}_2 - \delta \dot{\mathbf{g}}_1 - \boldsymbol{\omega}_2 \times (\delta \mathbf{g}_2 - \delta \mathbf{g}_1)). \end{aligned} \quad (9.229)$$

The internal moment $\tilde{\mathbf{M}}$ is computed as function of the relative rotation and velocity

$$\tilde{\mathbf{M}} = \tilde{\mathbf{M}}(\boldsymbol{\theta}, \boldsymbol{\omega}). \quad (9.230)$$

The nodal moments result from the VWP as

$$\begin{aligned} \delta \mathcal{L} &= \boldsymbol{\theta}_\delta^T \tilde{\mathbf{M}} \\ &= (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta})^T \mathbf{R}_{1h} \tilde{\mathbf{M}}, \end{aligned} \quad (9.231)$$

which states that the internal moment is applied to each node after pre-multiplication by \mathbf{R}_{1h}

$$\mathbf{M}_i = (-1)^i \mathbf{R}_{1h} \tilde{\mathbf{M}}(\boldsymbol{\theta}, \boldsymbol{\omega}). \quad (9.232)$$

Its linearization yields

$$\delta \mathbf{M}_i = (-1)^i \mathbf{R}_{1h} \left(\tilde{\mathbf{M}}_{/\boldsymbol{\theta}} \delta \boldsymbol{\theta} + \tilde{\mathbf{M}}_{/\boldsymbol{\omega}} \delta \boldsymbol{\omega} \right) - \mathbf{M}_i \times \boldsymbol{\theta}_{1\delta} \quad (9.233)$$

The complete linearized problem is

$$\begin{aligned} &\left[\begin{array}{cc} \mathbf{M}_{/\boldsymbol{\omega}} & -\mathbf{M}_{/\boldsymbol{\omega}} \\ -\mathbf{M}_{/\boldsymbol{\omega}} & \mathbf{M}_{/\boldsymbol{\omega}} \end{array} \right] \left\{ \begin{array}{c} \delta \boldsymbol{\omega}_1 \\ \delta \boldsymbol{\omega}_2 \end{array} \right\} \\ &+ \left[\begin{array}{cc} -\mathbf{M}_{/\boldsymbol{\omega}} (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times & \mathbf{0} \\ \mathbf{M}_{/\boldsymbol{\omega}} (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \boldsymbol{\theta}_{1\delta} \\ \boldsymbol{\theta}_{2\delta} \end{array} \right\} \\ &+ \left[\begin{array}{cc} \mathbf{M}_{/\boldsymbol{\theta}} & -\mathbf{M}_{/\boldsymbol{\theta}} \\ -\mathbf{M}_{/\boldsymbol{\theta}} & \mathbf{M}_{/\boldsymbol{\theta}} \end{array} \right] \left\{ \begin{array}{c} \boldsymbol{\theta}_{1\delta} \\ \boldsymbol{\theta}_{2\delta} \end{array} \right\} \\ &+ \left[\begin{array}{cc} \mathbf{M} \times & \mathbf{0} \\ -\mathbf{M} \times & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \boldsymbol{\theta}_{1\delta} \\ \boldsymbol{\theta}_{2\delta} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{M} \\ -\mathbf{M} \end{array} \right\}, \end{aligned} \quad (9.234)$$

where

$$\mathbf{M}_{/\boldsymbol{\theta}} = \mathbf{R}_{1h} \tilde{\mathbf{M}}_{/\boldsymbol{\theta}} \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1h}^T \quad (9.235a)$$

$$\mathbf{M}_{/\boldsymbol{\omega}} = \mathbf{R}_{1h} \tilde{\mathbf{M}}_{/\boldsymbol{\omega}} \mathbf{R}_{1h}^T. \quad (9.235b)$$

According to the simplifications of the updated-updated approach, the linearization becomes

$$\begin{aligned}
& \left[\begin{array}{cc} M_{/\omega} & -M_{/\omega} \\ -M_{/\omega} & M_{/\omega} \end{array} \right] \left\{ \begin{array}{c} \delta \dot{g}_1 \\ \delta \dot{g}_2 \end{array} \right\} \\
& + \left[\begin{array}{cc} -M_{/\omega} \omega_2 \times & M_{/\omega} \omega_2 \times \\ M_{/\omega} \omega_2 \times & -M_{/\omega} \omega_2 \times \end{array} \right] \left\{ \begin{array}{c} \delta g_1 \\ \delta g_2 \end{array} \right\} \\
& + \left[\begin{array}{cc} M_{/\theta} & -M_{/\theta} \\ -M_{/\theta} & M_{/\theta} \end{array} \right] \left\{ \begin{array}{c} \delta g_1 \\ \delta g_2 \end{array} \right\} \\
& + \left[\begin{array}{cc} M \times & \mathbf{0} \\ -M \times & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \delta g_1 \\ \delta g_2 \end{array} \right\} \stackrel{\text{uu}}{=} \left\{ \begin{array}{c} M \\ -M \end{array} \right\}. \tag{9.236}
\end{aligned}$$

Invariant Deformable Hinge

The rotation $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}/2$ is used to define an intermediate reference frame for the joint, whose orientation with respect to node 1 is

$$\begin{aligned}
\tilde{\mathbf{R}} &= \exp \left(\frac{1}{2} \exp^{-1} (\mathbf{R}_{1h}^T \mathbf{R}_{2h}) \right) \\
&= \exp \left(\tilde{\boldsymbol{\theta}} \times \right) \tag{9.237}
\end{aligned}$$

The orientation of the intermediate reference frame with respect to node 2 is defined by $-\tilde{\boldsymbol{\theta}}$, i.e. $\tilde{\mathbf{R}}^T$. The orientation of the intermediate frame with respect to the global frame is thus

$$\hat{\mathbf{R}} = \mathbf{R}_{1h} \tilde{\mathbf{R}} \tag{9.238a}$$

$$= \mathbf{R}_{2h} \tilde{\mathbf{R}}^T \tag{9.238b}$$

The perturbation of $\tilde{\mathbf{R}}$ is

$$\delta \tilde{\mathbf{R}} = \tilde{\boldsymbol{\theta}}_\delta \times \tilde{\mathbf{R}} \tag{9.239}$$

where

$$\delta \tilde{\boldsymbol{\theta}} = \frac{1}{2} \delta \boldsymbol{\theta} \tag{9.240}$$

and thus

$$\tilde{\boldsymbol{\theta}}_\delta = \boldsymbol{\Gamma}(\tilde{\boldsymbol{\theta}}) \frac{1}{2} \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \boldsymbol{\theta}_\delta \tag{9.241}$$

Another interesting result is obtained by considering that the perturbation of the relative orientation matrix must be equal to the perturbation of the square of the half-relative orientation, namely

$$\begin{aligned}
\boldsymbol{\theta}_\delta \times &= \delta (\tilde{\mathbf{R}} \tilde{\mathbf{R}}) (\tilde{\mathbf{R}} \tilde{\mathbf{R}})^T \\
&= \tilde{\boldsymbol{\theta}}_\delta \times + \tilde{\mathbf{R}} \tilde{\boldsymbol{\theta}}_\delta \times \tilde{\mathbf{R}}^T \tag{9.242}
\end{aligned}$$

and thus

$$\boldsymbol{\theta}_\delta = (\mathbf{I} + \tilde{\mathbf{R}}) \tilde{\boldsymbol{\theta}}_\delta \tag{9.243}$$

The matrix $(\mathbf{I} + \tilde{\mathbf{R}})$ has very interesting properties.

Property 1.

$$(\mathbf{I} + \tilde{\mathbf{R}}) \tilde{\mathbf{R}}^T = (\mathbf{I} + \tilde{\mathbf{R}}^T) \quad (9.244)$$

Property 2.

$$\tilde{\mathbf{R}}^T (\mathbf{I} + \tilde{\mathbf{R}}) = (\mathbf{I} + \tilde{\mathbf{R}}^T) \quad (9.245)$$

Property 3.

$$(\mathbf{I} + \tilde{\mathbf{R}}^T) \tilde{\mathbf{R}} = (\mathbf{I} + \tilde{\mathbf{R}}) \quad (9.246)$$

Property 4.

$$\tilde{\mathbf{R}} (\mathbf{I} + \tilde{\mathbf{R}}^T) = (\mathbf{I} + \tilde{\mathbf{R}}) \quad (9.247)$$

Property 5. As a consequence of Equations (9.244, 9.247),

$$\tilde{\mathbf{R}} (\mathbf{I} + \tilde{\mathbf{R}}) \tilde{\mathbf{R}}^T = (\mathbf{I} + \tilde{\mathbf{R}}) \quad (9.248)$$

which basically means that the transformation $(\mathbf{I} + \tilde{\mathbf{R}})$ is coaxial with the rotation described by matrix $\tilde{\mathbf{R}}$. A similar property exists for matrix $(\mathbf{I} + \tilde{\mathbf{R}}^T)$ (e.g. by transposing Equation (9.248)), while the following

Property 6.

$$\tilde{\mathbf{R}}^T (\mathbf{I} + \tilde{\mathbf{R}}) \tilde{\mathbf{R}} = (\mathbf{I} + \tilde{\mathbf{R}}) \quad (9.249)$$

is true as a consequence of Equations (9.245, 9.246).

Property 7. Another interesting property is

$$\mathbf{I} - (\mathbf{I} + \tilde{\mathbf{R}})^{-1} = (\mathbf{I} + \tilde{\mathbf{R}}^T)^{-1}. \quad (9.250)$$

This can be easily proved by considering that, according to Equation (9.247),

$$\begin{aligned} (\mathbf{I} + \tilde{\mathbf{R}})^{-1} &= (\tilde{\mathbf{R}} (\mathbf{I} + \tilde{\mathbf{R}}^T))^{-1} \\ &= (\mathbf{I} + \tilde{\mathbf{R}}^T)^{-1} \tilde{\mathbf{R}}^T, \end{aligned} \quad (9.251)$$

and, as a consequence,

$$\mathbf{I} = (\mathbf{I} + \tilde{\mathbf{R}}^T)^{-1} (\mathbf{I} + \tilde{\mathbf{R}}^T) \quad (9.252)$$

which reduces again to the identity matrix.

Property 8.

$$(\mathbf{I} + \tilde{\mathbf{R}})^T = (\mathbf{I} + \tilde{\mathbf{R}}^T) \quad (9.253)$$

Property 9.

$$(\mathbf{I} + \tilde{\mathbf{R}})^{-T} = (\mathbf{I} + \tilde{\mathbf{R}}^T)^{-1} \quad (9.254)$$

Property 10. As a consequence of Equations (9.244–9.247),

$$\hat{\mathbf{R}}(\mathbf{I} + \tilde{\mathbf{R}}) = \mathbf{R}_{1h}(\mathbf{I} + \tilde{\mathbf{R}}^T) \quad (9.255)$$

and

$$\hat{\mathbf{R}}(\mathbf{I} + \tilde{\mathbf{R}}^T)^{-1} = \mathbf{R}_{1h}(\mathbf{I} + \tilde{\mathbf{R}})^{-1} \quad (9.256)$$

and all other combinations.

It is convenient to define

$$\hat{\mathbf{I}} = \hat{\mathbf{R}}(\mathbf{I} + \tilde{\mathbf{R}})^{-1}\hat{\mathbf{R}}^T. \quad (9.257)$$

According to Equations (9.243) and (9.227), the perturbation of the intermediate relative rotation vector yields

$$\begin{aligned} \tilde{\boldsymbol{\theta}}_\delta &= (\mathbf{I} + \tilde{\mathbf{R}})^{-1} \boldsymbol{\theta}_\delta \\ &= (\mathbf{I} + \tilde{\mathbf{R}})^{-1} \mathbf{R}_{1h}^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) \\ &= (\mathbf{I} + \tilde{\mathbf{R}}^T)^{-1} \hat{\mathbf{R}}^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}). \end{aligned} \quad (9.258)$$

The perturbation of $\hat{\mathbf{R}}$, according to Equations (9.238a) and (9.238b), respectively yields

$$\begin{aligned} \delta \hat{\mathbf{R}} \hat{\mathbf{R}}^T &= \boldsymbol{\theta}_{1\delta} \times + \mathbf{R}_{1h} \tilde{\boldsymbol{\theta}}_\delta \times \mathbf{R}_{1h}^T \\ &= \boldsymbol{\theta}_{2\delta} \times - \hat{\mathbf{R}} \tilde{\boldsymbol{\theta}}_\delta \times \hat{\mathbf{R}}^T, \end{aligned} \quad (9.259)$$

so

$$\begin{aligned} \hat{\boldsymbol{\theta}}_\delta &= \boldsymbol{\theta}_{1\delta} + \mathbf{R}_{1h} \tilde{\boldsymbol{\theta}}_\delta && \text{Eq. (9.238a)} \\ &= \boldsymbol{\theta}_{1\delta} + \mathbf{R}_{1h}(\mathbf{I} + \tilde{\mathbf{R}})^{-1} \mathbf{R}_{1h}^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) \\ &= \boldsymbol{\theta}_{1\delta} + \hat{\mathbf{R}}(\mathbf{I} + \tilde{\mathbf{R}})^{-1} \hat{\mathbf{R}}^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) \\ &= \boldsymbol{\theta}_{2\delta} - \hat{\mathbf{R}} \tilde{\boldsymbol{\theta}}_\delta && \text{Eq. (9.238b)} \\ &= \boldsymbol{\theta}_{2\delta} - \hat{\mathbf{R}}(\mathbf{I} + \tilde{\mathbf{R}})^{-1} \mathbf{R}_{1h}^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) \\ &= \boldsymbol{\theta}_{2\delta} - \hat{\mathbf{R}}(\mathbf{I} + \tilde{\mathbf{R}}^T)^{-1} \hat{\mathbf{R}}^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) \\ &= \hat{\mathbf{I}} \boldsymbol{\theta}_{2\delta} + \hat{\mathbf{I}}^T \boldsymbol{\theta}_{1\delta} \end{aligned} \quad (9.260)$$

The perturbation of matrix $\hat{\mathbf{I}}$ yields

$$\begin{aligned}\delta\hat{\mathbf{I}} &= \delta\hat{\mathbf{R}}\left(\mathbf{I} + \tilde{\mathbf{R}}\right)^{-1}\hat{\mathbf{R}}^T \\ &\quad + \hat{\mathbf{R}}\delta\left(\mathbf{I} + \tilde{\mathbf{R}}\right)^{-1}\hat{\mathbf{R}}^T \\ &\quad + \hat{\mathbf{R}}\left(\mathbf{I} + \tilde{\mathbf{R}}\right)^{-1}\delta\hat{\mathbf{R}}^T\end{aligned}\tag{9.261a}$$

$$\begin{aligned}&= \hat{\boldsymbol{\theta}}_\delta \times \hat{\mathbf{I}} - \hat{\mathbf{R}}\left(\mathbf{I} + \tilde{\mathbf{R}}\right)^{-1}\tilde{\boldsymbol{\theta}}_\delta \times \tilde{\mathbf{R}}\left(\mathbf{I} + \tilde{\mathbf{R}}\right)^{-1}\hat{\mathbf{R}}^T + \hat{\mathbf{I}}\hat{\boldsymbol{\theta}}_\delta \times {}^T \\ &= \hat{\boldsymbol{\theta}}_\delta \times \hat{\mathbf{I}} - \hat{\mathbf{R}}\left(\mathbf{I} + \tilde{\mathbf{R}}\right)^{-1}\underbrace{\mathbf{R}_{1h}^T\mathbf{R}_{1h}}_I\tilde{\boldsymbol{\theta}}_\delta \times \underbrace{\mathbf{R}_{1h}^T\mathbf{R}_{1h}}_I\tilde{\mathbf{R}}\left(\mathbf{I} + \tilde{\mathbf{R}}\right)^{-1}\hat{\mathbf{R}}^T - \hat{\mathbf{I}}\hat{\boldsymbol{\theta}}_\delta \times \\ &= \hat{\boldsymbol{\theta}}_\delta \times \hat{\mathbf{I}} - \underbrace{\hat{\mathbf{R}}\left(\mathbf{I} + \tilde{\mathbf{R}}\right)^{-1}\mathbf{R}_{1h}^T}_{\hat{\mathbf{I}}^T}\underbrace{\mathbf{R}_{1h}\tilde{\boldsymbol{\theta}}_\delta \times \mathbf{R}_{1h}^T}_{(\mathbf{R}_{1h}\tilde{\boldsymbol{\theta}}_\delta) \times}\underbrace{\hat{\mathbf{R}}\left(\mathbf{I} + \tilde{\mathbf{R}}\right)^{-1}\hat{\mathbf{R}}^T}_{\hat{\mathbf{I}}}-\hat{\mathbf{I}}\hat{\boldsymbol{\theta}}_\delta \times \\ &= \hat{\boldsymbol{\theta}}_\delta \times \hat{\mathbf{I}} - \hat{\mathbf{I}}^T\left(\underbrace{\mathbf{R}_{1h}\left(\mathbf{I} + \tilde{\mathbf{R}}\right)^{-1}\mathbf{R}_{1h}^T}_{\hat{\mathbf{I}}}(\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta})\right) \times \hat{\mathbf{I}} - \hat{\mathbf{I}}\hat{\boldsymbol{\theta}}_\delta \times \\ &= (\hat{\mathbf{I}}\boldsymbol{\theta}_{2\delta} + \hat{\mathbf{I}}^T\boldsymbol{\theta}_{1\delta}) \times \hat{\mathbf{I}} - \hat{\mathbf{I}}^T(\hat{\mathbf{I}}(\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta})) \times \hat{\mathbf{I}} - \hat{\mathbf{I}}(\hat{\mathbf{I}}\boldsymbol{\theta}_{2\delta} + \hat{\mathbf{I}}^T\boldsymbol{\theta}_{1\delta}) \times\end{aligned}\tag{9.261b}$$

The perturbation of matrix $\hat{\mathbf{I}}$ is generally useful when multiplying a generic vector \mathbf{v} , resulting in

$$\begin{aligned}\delta\hat{\mathbf{I}}\mathbf{v} &= \left(\left(\hat{\mathbf{I}}\mathbf{v} \times - (\hat{\mathbf{I}}\mathbf{v}) \times\right)\hat{\mathbf{I}}^T - \hat{\mathbf{I}}^T(\hat{\mathbf{I}}\mathbf{v}) \times \hat{\mathbf{I}}\right)\boldsymbol{\theta}_{1\delta} \\ &\quad + \left(\left(\hat{\mathbf{I}}\mathbf{v} \times - (\hat{\mathbf{I}}\mathbf{v}) \times\right)\hat{\mathbf{I}} + \hat{\mathbf{I}}^T(\hat{\mathbf{I}}\mathbf{v}) \times \hat{\mathbf{I}}\right)\boldsymbol{\theta}_{2\delta} \\ &= \hat{\mathbf{I}}_{1(v)}\boldsymbol{\theta}_{1\delta} + \hat{\mathbf{I}}_{2(v)}\boldsymbol{\theta}_{2\delta}\end{aligned}\tag{9.262a}$$

$$\begin{aligned}\delta(\hat{\mathbf{I}}^T)\mathbf{v} &= \left(\left(\hat{\mathbf{I}}^T\mathbf{v} \times - (\hat{\mathbf{I}}^T\mathbf{v}) \times\right)\hat{\mathbf{I}}^T + \hat{\mathbf{I}}^T(\hat{\mathbf{I}}\mathbf{v}) \times \hat{\mathbf{I}}\right)\boldsymbol{\theta}_{1\delta} \\ &\quad + \left(\left(\hat{\mathbf{I}}^T\mathbf{v} \times - (\hat{\mathbf{I}}^T\mathbf{v}) \times\right)\hat{\mathbf{I}} - \hat{\mathbf{I}}^T(\hat{\mathbf{I}}\mathbf{v}) \times \hat{\mathbf{I}}\right)\boldsymbol{\theta}_{2\delta} \\ &= (\hat{\mathbf{I}}^T)_{1(v)}\boldsymbol{\theta}_{1\delta} + (\hat{\mathbf{I}}^T)_{2(v)}\boldsymbol{\theta}_{2\delta}.\end{aligned}\tag{9.262b}$$

Relative Rotation Vector. The relative rotation vector is $\boldsymbol{\theta}$; by definition, this does not depend on the reference frame that is considered, among those intermediate between node 1 and 2, since it is coaxial to any rotation intermediate between the orientations of the two nodes; in detail

$$\tilde{\mathbf{R}}^T\boldsymbol{\theta} = \boldsymbol{\theta},\tag{9.263}$$

since $\tilde{\boldsymbol{\theta}}$ is co-axial to $\boldsymbol{\theta}$ by construction.

This implies that

$$\begin{aligned}
\delta\boldsymbol{\theta} &= \delta(\tilde{\mathbf{R}}^T \boldsymbol{\theta}) \\
&= \tilde{\mathbf{R}}^T (\boldsymbol{\theta} \times \tilde{\boldsymbol{\theta}}_\delta + \delta\boldsymbol{\theta}) \\
&= \tilde{\mathbf{R}}^T (\boldsymbol{\theta} \times \Gamma(\tilde{\boldsymbol{\theta}}) \delta\tilde{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}) \\
&= \tilde{\mathbf{R}}^T \left(2\tilde{\boldsymbol{\theta}} \times \Gamma(\tilde{\boldsymbol{\theta}}) \frac{\delta\boldsymbol{\theta}}{2} + \delta\boldsymbol{\theta} \right) \\
&= \tilde{\mathbf{R}}^T (\tilde{\boldsymbol{\theta}} \times \Gamma(\tilde{\boldsymbol{\theta}}) + \mathbf{I}) \delta\boldsymbol{\theta} \\
&= \delta\boldsymbol{\theta},
\end{aligned} \tag{9.264}$$

since, by definition,

$$\mathbf{I} + \tilde{\boldsymbol{\theta}} \times \Gamma(\tilde{\boldsymbol{\theta}}) = \tilde{\mathbf{R}}. \tag{9.265}$$

The perturbation of the relative rotation vector $\boldsymbol{\theta}$,

$$\delta\bar{\boldsymbol{\theta}} = \delta\boldsymbol{\theta}, \tag{9.266}$$

is used as perturbation of the measure of the straining of the joint.

Relative Angular Velocity. The relative angular velocity between the two nodes, in the intermediate reference frame $\hat{\mathbf{R}}$, is

$$\bar{\boldsymbol{\omega}} = \hat{\mathbf{R}}^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1), \tag{9.267}$$

which corresponds to that of the attached case, Eq. (9.226), but projected in the material frame by $\hat{\mathbf{R}}^T$ instead of the reference frame of node 1, \mathbf{R}_{1h} . The linearization of the angular velocity of Equation (9.267)

yields

$$\begin{aligned}
\delta\bar{\omega} &= \hat{\mathbf{R}}^T \left((\omega_2 - \omega_1) \times (\theta_{1\delta} + \mathbf{R}_{1h}\tilde{\theta}_\delta) + \delta\omega_2 - \delta\omega_1 \right) \\
&= \hat{\mathbf{R}}^T (\delta\omega_2 - \delta\omega_1) \\
&\quad + \hat{\mathbf{R}}^T (\omega_2 - \omega_1) \times \mathbf{R}_{1h} \left(\mathbf{I} + \tilde{\mathbf{R}} \right)^{-1} \mathbf{R}_{1h}^T \theta_{2\delta} \\
&\quad + \hat{\mathbf{R}}^T (\omega_2 - \omega_1) \times \left(\mathbf{I} - \mathbf{R}_{1h} \left(\mathbf{I} + \tilde{\mathbf{R}} \right)^{-1} \mathbf{R}_{1h}^T \right) \theta_{1\delta} \\
&= \hat{\mathbf{R}}^T (\delta\omega_2 - \delta\omega_1) \\
&\quad + \hat{\mathbf{R}}^T (\omega_2 - \omega_1) \times \hat{\mathbf{R}} \left(\mathbf{I} + \tilde{\mathbf{R}} \right)^{-1} \hat{\mathbf{R}}^T \theta_{2\delta} \\
&\quad + \hat{\mathbf{R}}^T (\omega_2 - \omega_1) \times \hat{\mathbf{R}} \left(\mathbf{I} - \left(\mathbf{I} + \tilde{\mathbf{R}} \right)^{-1} \right) \hat{\mathbf{R}}^T \theta_{1\delta} \\
&= \hat{\mathbf{R}}^T (\delta\omega_2 - \delta\omega_1) \\
&\quad + \bar{\omega} \times \left(\left(\mathbf{I} + \tilde{\mathbf{R}} \right)^{-1} \hat{\mathbf{R}}^T \theta_{2\delta} + \left(\mathbf{I} - \left(\mathbf{I} + \tilde{\mathbf{R}} \right)^{-1} \right) \hat{\mathbf{R}}^T \theta_{1\delta} \right) \\
&= \hat{\mathbf{R}}^T (\delta\omega_2 - \delta\omega_1) \\
&\quad + \bar{\omega} \times \left(\left(\mathbf{I} + \tilde{\mathbf{R}} \right)^{-1} \hat{\mathbf{R}}^T \theta_{2\delta} + \left(\mathbf{I} + \tilde{\mathbf{R}}^T \right)^{-1} \hat{\mathbf{R}}^T \theta_{1\delta} \right) \\
&= \hat{\mathbf{R}}^T \left(\delta\omega_2 - \delta\omega_1 + (\omega_2 - \omega_1) \times \left(\hat{\mathbf{I}}\theta_{2\delta} + \hat{\mathbf{I}}^T\theta_{1\delta} \right) \right) \tag{9.268}
\end{aligned}$$

since

$$\tilde{\mathbf{R}} \left(\mathbf{I} + \tilde{\mathbf{R}} \right) \tilde{\mathbf{R}}^T = \left(\mathbf{I} + \tilde{\mathbf{R}} \right) \tag{9.269}$$

and thus

$$\begin{aligned}
\left(\mathbf{I} + \tilde{\mathbf{R}} \right)^{-1} &= \tilde{\mathbf{R}}^{-T} \left(\mathbf{I} + \tilde{\mathbf{R}} \right)^{-1} \tilde{\mathbf{R}}^{-1} \\
&= \tilde{\mathbf{R}} \left(\mathbf{I} + \tilde{\mathbf{R}} \right)^{-1} \tilde{\mathbf{R}}^T,
\end{aligned} \tag{9.270}$$

where Equation (9.250) has been used.

Intermediate Angular Velocity. Consider now

$$\begin{aligned}
\hat{\omega} \times &= \dot{\hat{\mathbf{R}}} \hat{\mathbf{R}}^T \\
&= \omega_1 \times + \mathbf{R}_{1h} \tilde{\omega} \times \mathbf{R}_{1h}^T \\
&= \omega_2 \times - \hat{\mathbf{R}} \tilde{\omega} \times \hat{\mathbf{R}}^T,
\end{aligned} \tag{9.271}$$

where $\tilde{\omega}$ is the angular velocity associated to the differentiation of matrix $\tilde{\mathbf{R}}$ with respect to time:

$$\tilde{\omega} = \dot{\tilde{\mathbf{R}}} \tilde{\mathbf{R}}^T. \tag{9.272}$$

A relationship between that derivative and the remaining angular velocities results from

$$\begin{aligned}
\omega \times &= \frac{d}{dt} \left(\tilde{\mathbf{R}} \tilde{\mathbf{R}}^T \right) \left(\tilde{\mathbf{R}} \tilde{\mathbf{R}}^T \right)^T \\
&= \tilde{\omega} \times + \tilde{\mathbf{R}} \tilde{\omega} \times \tilde{\mathbf{R}}^T,
\end{aligned} \tag{9.273}$$

so

$$\boldsymbol{\omega} = (\mathbf{I} + \tilde{\mathbf{R}}) \tilde{\boldsymbol{\omega}}; \quad (9.274)$$

as a consequence,

$$\tilde{\boldsymbol{\omega}} = (\mathbf{I} + \tilde{\mathbf{R}})^{-1} \mathbf{R}_{1h}^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1). \quad (9.275)$$

The absolute velocity of the intermediate orientation results in

$$\hat{\boldsymbol{\omega}} = \hat{\mathbf{I}}\boldsymbol{\omega}_2 + \hat{\mathbf{I}}^T\boldsymbol{\omega}_1; \quad (9.276)$$

its perturbation becomes

$$\begin{aligned} \delta\hat{\boldsymbol{\omega}} &= \delta(\hat{\mathbf{I}}\boldsymbol{\omega}_2 + \hat{\mathbf{I}}^T\boldsymbol{\omega}_1) \\ &= \hat{\mathbf{I}}\delta\boldsymbol{\omega}_2 + \hat{\mathbf{I}}^T\delta\boldsymbol{\omega}_1 \\ &+ \left(\begin{array}{c} (\hat{\mathbf{I}}\boldsymbol{\omega}_2 \times + \hat{\mathbf{I}}^T\boldsymbol{\omega}_1 \times) \hat{\mathbf{I}} \\ -(\hat{\mathbf{I}}\boldsymbol{\omega}_2 + \hat{\mathbf{I}}^T\boldsymbol{\omega}_1) \times \hat{\mathbf{I}} \\ + \hat{\mathbf{I}}(\hat{\mathbf{I}}^T(\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1)) \times \hat{\mathbf{I}}^T \end{array} \right) \theta_{2\delta} \\ &+ \left(\begin{array}{c} (\hat{\mathbf{I}}\boldsymbol{\omega}_2 \times + \hat{\mathbf{I}}^T\boldsymbol{\omega}_1 \times) \hat{\mathbf{I}}^T \\ -(\hat{\mathbf{I}}\boldsymbol{\omega}_2 + \hat{\mathbf{I}}^T\boldsymbol{\omega}_1) \times \hat{\mathbf{I}}^T \\ - \hat{\mathbf{I}}(\hat{\mathbf{I}}^T(\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1)) \times \hat{\mathbf{I}}^T \end{array} \right) \theta_{1\delta} \\ &= \hat{\mathbf{I}}\delta\boldsymbol{\omega}_2 + \hat{\mathbf{I}}^T\delta\boldsymbol{\omega}_1 + \hat{\mathbf{I}}_2\theta_{2\delta} + \hat{\mathbf{I}}_1\theta_{1\delta} \\ &\stackrel{\text{uu}}{=} \hat{\mathbf{I}}\delta\dot{\mathbf{g}}_2 + \hat{\mathbf{I}}^T\delta\dot{\mathbf{g}}_1 + (\hat{\mathbf{I}}_2 - \hat{\mathbf{I}}\boldsymbol{\omega}_2 \times) \delta\mathbf{g}_2 + (\hat{\mathbf{I}}_1 - \hat{\mathbf{I}}^T\boldsymbol{\omega}_1 \times) \delta\mathbf{g}_1 \end{aligned} \quad (9.277)$$

Equilibrium Equations. The nodal moment results from the VWP, considering the perturbation of the relative orientation $\bar{\boldsymbol{\theta}}_\delta$,

$$\begin{aligned} \delta\mathcal{L} &= \bar{\boldsymbol{\theta}}_\delta^T \tilde{\mathbf{M}} \\ &= (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta})^T \hat{\mathbf{R}} \tilde{\mathbf{M}}, \end{aligned} \quad (9.278)$$

which corresponds to

$$\mathbf{M}_i = (-1)^i \hat{\mathbf{R}} \tilde{\mathbf{M}} (\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\omega}}). \quad (9.279)$$

It depends on the relative rotation vector $\bar{\boldsymbol{\theta}} = \hat{\mathbf{R}}^T \boldsymbol{\theta} = \boldsymbol{\theta}$, and on the relative angular velocity $\bar{\boldsymbol{\omega}}$, illustrated by Equation (9.267), expressed in the intermediate reference frame.

Equilibrium Perturbation. The perturbation of the moment then yields

$$\begin{aligned} \delta\mathbf{M}_i &= (-1)^i \hat{\mathbf{R}} (\tilde{\mathbf{M}}_{/\boldsymbol{\theta}} \delta\bar{\boldsymbol{\theta}} + \tilde{\mathbf{M}}_{/\boldsymbol{\omega}} \delta\bar{\boldsymbol{\omega}}) \\ &- \mathbf{M}_i \times \hat{\mathbf{R}} (\mathbf{I} + \tilde{\mathbf{R}})^{-1} \hat{\mathbf{R}}^T \boldsymbol{\theta}_{2\delta} \\ &- \mathbf{M}_i \times \hat{\mathbf{R}} (\mathbf{I} + \tilde{\mathbf{R}}^T)^{-1} \hat{\mathbf{R}}^T \boldsymbol{\theta}_{1\delta} \\ &= (-1)^i \hat{\mathbf{R}} (\tilde{\mathbf{M}}_{/\boldsymbol{\theta}} \delta\bar{\boldsymbol{\theta}} + \tilde{\mathbf{M}}_{/\boldsymbol{\omega}} \delta\bar{\boldsymbol{\omega}}) - \mathbf{M}_i \times (\hat{\mathbf{I}}\boldsymbol{\theta}_{2\delta} + \hat{\mathbf{I}}^T\boldsymbol{\theta}_{1\delta}). \end{aligned} \quad (9.280)$$

The complete linearized problem is

$$\begin{aligned}
& \left[\begin{array}{cc} M_{/\omega} & -M_{/\omega} \\ -M_{/\omega} & M_{/\omega} \end{array} \right] \left\{ \begin{array}{c} \delta\omega_1 \\ \delta\omega_2 \end{array} \right\} \\
& + \left[\begin{array}{cc} -M_{/\omega}(\omega_2 - \omega_1) \times \hat{\mathbf{I}}^T & -M_{/\omega}(\omega_2 - \omega_1) \times \hat{\mathbf{I}} \\ M_{/\omega}(\omega_2 - \omega_1) \times \hat{\mathbf{I}}^T & M_{/\omega}(\omega_2 - \omega_1) \times \hat{\mathbf{I}} \end{array} \right] \left\{ \begin{array}{c} \theta_{1\delta} \\ \theta_{2\delta} \end{array} \right\} \\
& \quad + \left[\begin{array}{cc} M_{/\theta} & -M_{/\theta} \\ -M_{/\theta} & M_{/\theta} \end{array} \right] \left\{ \begin{array}{c} \theta_{1\delta} \\ \theta_{2\delta} \end{array} \right\} \\
& + \left[\begin{array}{cc} \mathbf{M} \times \hat{\mathbf{I}}^T & \mathbf{M} \times \hat{\mathbf{I}} \\ -\mathbf{M} \times \hat{\mathbf{I}}^T & -\mathbf{M} \times \hat{\mathbf{I}} \end{array} \right] \left\{ \begin{array}{c} \theta_{1\delta} \\ \theta_{2\delta} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{M} \\ -\mathbf{M} \end{array} \right\} \tag{9.281}
\end{aligned}$$

where

$$\mathbf{M}_{/\theta} = \hat{\mathbf{R}} \tilde{\mathbf{M}}_{/\theta} \Gamma(\theta)^{-1} \mathbf{R}_{1h}^T \tag{9.282a}$$

$$\mathbf{M}_{/\omega} = \hat{\mathbf{R}} \tilde{\mathbf{M}}_{/\omega} \hat{\mathbf{R}}^T. \tag{9.282b}$$

Accounting for the simplifications of the updated-updated approach, it becomes

$$\begin{aligned}
& \left[\begin{array}{cc} M_{/\omega} & -M_{/\omega} \\ -M_{/\omega} & M_{/\omega} \end{array} \right] \left\{ \begin{array}{c} \delta\dot{\mathbf{g}}_1 \\ \delta\dot{\mathbf{g}}_2 \end{array} \right\} \\
& + \left[\begin{array}{cc} -M_{/\omega}(\omega_2 \times \hat{\mathbf{I}}^T + \omega_1 \times \hat{\mathbf{I}}) & M_{/\omega}(\omega_2 \times \hat{\mathbf{I}}^T + \omega_1 \times \hat{\mathbf{I}}) \\ M_{/\omega}(\omega_2 \times \hat{\mathbf{I}}^T + \omega_1 \times \hat{\mathbf{I}}) & -M_{/\omega}(\omega_2 \times \hat{\mathbf{I}}^T + \omega_1 \times \hat{\mathbf{I}}) \end{array} \right] \left\{ \begin{array}{c} \delta\mathbf{g}_1 \\ \delta\mathbf{g}_2 \end{array} \right\} \\
& \quad + \left[\begin{array}{cc} M_{/\theta} & -M_{/\theta} \\ -M_{/\theta} & M_{/\theta} \end{array} \right] \left\{ \begin{array}{c} \delta\mathbf{g}_1 \\ \delta\mathbf{g}_2 \end{array} \right\} \\
& + \left[\begin{array}{cc} \mathbf{M} \times \hat{\mathbf{I}}^T & \mathbf{M} \times \hat{\mathbf{I}} \\ -\mathbf{M} \times \hat{\mathbf{I}}^T & -\mathbf{M} \times \hat{\mathbf{I}} \end{array} \right] \left\{ \begin{array}{c} \delta\mathbf{g}_1 \\ \delta\mathbf{g}_2 \end{array} \right\} \stackrel{\text{uu}}{=} \left\{ \begin{array}{c} \mathbf{M} \\ -\mathbf{M} \end{array} \right\} \tag{9.283}
\end{aligned}$$

Note on attached vs. invariant deformable hinge

The moment applied by the attached formulation to node 1, in the global reference frame, is

$$\mathbf{M}_a = \mathbf{R}_{1h} \tilde{\mathbf{M}}_a(\theta), \tag{9.284}$$

while the moment applied by the invariant formulation to node 1, in the global reference frame, is

$$\begin{aligned}
\mathbf{M}_i &= \hat{\mathbf{R}} \tilde{\mathbf{M}}_i(\theta) \\
&= \mathbf{R}_{1h} \hat{\mathbf{R}} \tilde{\mathbf{M}}_i(\hat{\mathbf{R}}^T \theta), \tag{9.285}
\end{aligned}$$

since $\hat{\mathbf{R}}^T \theta = \theta$. This means that whatever formula is used for the constitutive law, in the invariant case, it is equivalent to using an attached formula with a constitutive law re-oriented by $\hat{\mathbf{R}}$,

$$\tilde{\mathbf{M}}_a(\theta) = \hat{\mathbf{R}} \tilde{\mathbf{M}}_i(\hat{\mathbf{R}}^T \theta); \tag{9.286}$$

similar considerations apply to the viscous portion of the constitutive law. This transformation should be kept in mind when determining the properties of the deformable component. In fact, exchanging the locations where the constitutive law is evaluated implies a transformation of the constitutive matrix; in case a linear elastic constitutive law,

$$\tilde{\mathbf{M}}_i = \hat{\mathbf{K}}_i \theta, \tag{9.287}$$

with a constant (symmetric, positive definite) $\tilde{\mathbf{K}}_i$ matrix, is used for the invariant formula, which appears to be a natural solution for simple elastic hinges, it is equivalent to a nonlinear elastic constitutive law when transposing it into the attached formulation. The two formulas are coincident, and independent from $\boldsymbol{\theta}$, only in case of an isotropic spring, namely $\tilde{\mathbf{K}}_i = k\mathbf{I}$.

A “natural” solution, for a geometrically and materially symmetrical component, behaves the same when the order of the nodes is swapped. This corresponds to the “invariant” formula for the deformable hinge. However, typical experiments to determine the mechanical behavior of a component would rather consist in straining it while measuring the resulting loads at one or both ends, not in the intermediate location where the constitutive law would be naturally applied. As a consequence, a “natural” procedure for the determination of the constitutive law would consist in determining the $\tilde{\mathbf{M}}_a(\boldsymbol{\theta})$ law first; then, assuming the moment can be expressed in the form

$$\mathbf{R}_{1h}^T \mathbf{M}_a(\boldsymbol{\theta}) = \tilde{\mathbf{M}}_a(\boldsymbol{\theta}), \quad (9.288)$$

matrix $\tilde{\mathbf{R}}$ would be computed from the measured relative rotation of the extremities of the component, $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}/2$, and the constitutive law $\tilde{\mathbf{M}}_i(\boldsymbol{\theta})$ would be computed as

$$\tilde{\mathbf{M}}_i(\tilde{\mathbf{R}}^T \boldsymbol{\theta}) = \tilde{\mathbf{R}}^T \tilde{\mathbf{M}}_a(\boldsymbol{\theta}) \quad (9.289)$$

to yield the invariant constitutive law.

9.2.3 Deformable Displacement Joint

The deformable displacement joint applies an internal force to two nodes at a specified point that may be offset from the nodes. The force may depend on the relative position and velocity of the nodes at the point of application through a 3D constitutive law.

The points whose relative displacement represents the measure of the straining can be offset from the nodes by rigid offsets $\tilde{\mathbf{f}}_1$ and $\tilde{\mathbf{f}}_2$, which are defined in the reference frame of the respective nodes. So the offsets in the global reference frame are

$$\mathbf{f}_1 = \mathbf{R}_1 \tilde{\mathbf{f}}_1 \quad (9.290a)$$

$$\mathbf{f}_2 = \mathbf{R}_2 \tilde{\mathbf{f}}_2. \quad (9.290b)$$

The constitutive law is expressed in a reference frame that may be rotated from that of the node by rigid rotations $\tilde{\mathbf{R}}_{1h}$ and $\tilde{\mathbf{R}}_{2h}$. So the orientations of the constitutive law in the global reference frame are

$$\mathbf{R}_{1h} = \mathbf{R}_1 \tilde{\mathbf{R}}_{1h} \quad (9.291a)$$

$$\mathbf{R}_{2h} = \mathbf{R}_2 \tilde{\mathbf{R}}_{2h}. \quad (9.291b)$$

As for the **deformable hinge**, the constitutive law of this joint may be either attached to one node or defined in an intermediate reference frame that accounts for the relative orientation of the two nodes, so that the order in which the nodes are defined becomes irrelevant.

The relative position in the absolute reference frame, \mathbf{d} , is

$$\mathbf{d} = \mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1 - \mathbf{f}_1. \quad (9.292)$$

The relative velocity is

$$\dot{\mathbf{d}} = \dot{\mathbf{x}}_2 + \boldsymbol{\omega}_2 \times \mathbf{f}_2 - \dot{\mathbf{x}}_1 - \boldsymbol{\omega}_1 \times \mathbf{f}_1. \quad (9.293)$$

Attached Deformable Displacement Joint

The “attached” form of the deformable displacement joint is obtained by projecting the relative position \mathbf{d} in a reference frame attached to node 1:

$$\tilde{\mathbf{d}} = \mathbf{R}_{1h}^T \mathbf{d} \quad (9.294)$$

The relative velocity is

$$\dot{\tilde{\mathbf{d}}} = \mathbf{R}_{1h}^T (\dot{\mathbf{d}} - \boldsymbol{\omega}_1 \times \mathbf{d}), \quad (9.295)$$

but it can be replaced by

$$\dot{\tilde{\mathbf{d}}} = \mathbf{R}_{1h}^T (\dot{\mathbf{d}}_1 - \boldsymbol{\omega}_1 \times \mathbf{d}_1), \quad (9.296)$$

where

$$\mathbf{d}_1 = \mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1 \quad (9.297a)$$

$$\dot{\mathbf{d}}_1 = \dot{\mathbf{x}}_2 + \boldsymbol{\omega}_2 \times \mathbf{f}_2 - \dot{\mathbf{x}}_1, \quad (9.297b)$$

since the relative velocity associated to the rotation of \mathbf{f}_1 , the offset of the reference point attached to node 1, is zero by definition. The linearization of the distance yields

$$\begin{aligned} \delta\tilde{\mathbf{d}} &= \mathbf{R}_{1h}^T (\delta\mathbf{d} - \mathbf{d} \times \boldsymbol{\theta}_{1\delta}) \\ &= \mathbf{R}_{1h}^T (\delta\mathbf{x}_2 - \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} - \delta\mathbf{x}_1 + \mathbf{f}_1 \times \boldsymbol{\theta}_{1\delta}) + \mathbf{R}_{1h}^T \mathbf{d} \times \boldsymbol{\theta}_{1\delta} \\ &= \mathbf{R}_{1h}^T (\delta\mathbf{x}_2 - \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} - \delta\mathbf{x}_1 + (\mathbf{x}_2 + \mathbf{f}_2 - \mathbf{x}_1) \times \boldsymbol{\theta}_{\delta 1}) \\ &= \mathbf{R}_{1h}^T (\delta\mathbf{x}_2 - \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} - \delta\mathbf{x}_1 + \mathbf{d}_1 \times \boldsymbol{\theta}_{1\delta}) \\ &\stackrel{\text{uu}}{=} \mathbf{R}_{1h}^T (\delta\mathbf{x}_2 - \mathbf{f}_2 \times \delta\mathbf{g}_2 - \delta\mathbf{x}_1 + \mathbf{d}_1 \times \delta\mathbf{g}_1) \end{aligned} \quad (9.298)$$

while the linearization of the relative velocity yields

$$\begin{aligned} \delta\dot{\tilde{\mathbf{d}}} &= \mathbf{R}_{1h}^T (\delta\dot{\mathbf{d}} + \mathbf{d} \times \delta\boldsymbol{\omega}_1 - \boldsymbol{\omega}_1 \times \delta\mathbf{d} + (\dot{\mathbf{d}} - \boldsymbol{\omega}_1 \times \mathbf{d}) \times \boldsymbol{\theta}_{1\delta}) \\ &= \mathbf{R}_{1h}^T \left(\begin{array}{c} \delta\dot{\mathbf{x}}_2 - \mathbf{f}_2 \times \delta\boldsymbol{\omega}_2 - \delta\dot{\mathbf{x}}_1 + \mathbf{d}_1 \times \delta\boldsymbol{\omega}_1 \\ - \boldsymbol{\omega}_1 \times \delta\mathbf{x}_2 - (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} \\ + \boldsymbol{\omega}_1 \times \delta\mathbf{x}_1 + (\dot{\mathbf{d}} - \boldsymbol{\omega}_1 \times \mathbf{d}) \times \boldsymbol{\theta}_{1\delta} \end{array} \right) \\ &\stackrel{\text{uu}}{=} \mathbf{R}_{1h}^T \left(\begin{array}{c} \delta\dot{\mathbf{x}}_2 - \mathbf{f}_2 \times \delta\dot{\mathbf{g}}_2 - \delta\dot{\mathbf{x}}_1 + \mathbf{d}_1 \times \delta\dot{\mathbf{g}}_1 \\ - \boldsymbol{\omega}_1 \times \delta\mathbf{x}_2 - ((\boldsymbol{\omega}_2 \times \mathbf{f}_2) \times - \boldsymbol{\omega}_1 \times \mathbf{f}_2 \times) \delta\mathbf{g}_2 \\ + \boldsymbol{\omega}_1 \times \delta\mathbf{x}_1 + (\dot{\mathbf{d}}_1 \times - \boldsymbol{\omega}_1 \times \mathbf{d}_1 \times) \delta\mathbf{g}_1 \end{array} \right) \end{aligned} \quad (9.299)$$

The nodal forces and moments result from the VWP

$$\begin{aligned} \delta\mathcal{L} &= \delta\tilde{\mathbf{d}}^T \tilde{\mathbf{F}} \\ &= (\delta\mathbf{x}_2 - \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} - \delta\mathbf{x}_1 + \mathbf{d}_1 \times \boldsymbol{\theta}_{1\delta})^T \mathbf{R}_{1h} \tilde{\mathbf{F}}, \end{aligned} \quad (9.300)$$

which corresponds to

$$\mathbf{F}_i = (-1)^i \mathbf{R}_{1h} \tilde{\mathbf{F}} (\tilde{\mathbf{d}}, \dot{\tilde{\mathbf{d}}}) \quad (9.301a)$$

$$\mathbf{M}_i = \mathbf{d}_i \times \mathbf{F}_i \quad (9.301b)$$

where \mathbf{d}_1 has already been defined, and

$$\mathbf{d}_2 = \mathbf{f}_2; \quad (9.302)$$

note that

$$\begin{aligned} \boldsymbol{\omega}_2 \times \mathbf{f}_2 &= \boldsymbol{\omega}_2 \times \mathbf{d}_2 \\ &= \dot{\mathbf{d}}_2. \end{aligned} \quad (9.303)$$

Their linearization yields

$$\delta \mathbf{F}_i = (-1)^i \mathbf{R}_{1h} \left(\tilde{\mathbf{F}}_{/\dot{\mathbf{d}}} \delta \tilde{\mathbf{d}} + \tilde{\mathbf{F}}_{/\dot{\mathbf{d}}} \delta \dot{\mathbf{d}} \right) - \mathbf{F}_i \times \boldsymbol{\theta}_{1\delta} \quad (9.304a)$$

$$\delta \mathbf{M}_i = \mathbf{d}_i \times \delta \mathbf{F}_i - \mathbf{F}_i \times \delta \mathbf{d}_i \quad (9.304b)$$

The complete linearized problem, after applying the updated-updated approximation, is

$$\begin{aligned} & \left[\begin{array}{ccc} \mathbf{F}_{/\dot{\mathbf{d}}} & -\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_1 \times & -\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_2 \times \\ \mathbf{d}_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} & -\mathbf{d}_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_1 \times & -\mathbf{d}_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_2 \times \\ -\mathbf{F}_{/\dot{\mathbf{d}}} & \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_1 \times & \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_2 \times \\ -\mathbf{d}_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} & \mathbf{d}_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_1 \times & \mathbf{d}_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_2 \times \end{array} \right] \left\{ \begin{array}{l} \delta \dot{\mathbf{x}}_1 \\ \delta \dot{\mathbf{g}}_1 \\ \delta \dot{\mathbf{x}}_2 \\ \delta \dot{\mathbf{g}}_2 \end{array} \right\} \\ & + \left[\begin{array}{c} -\mathbf{F}_{/\dot{\mathbf{d}}} \boldsymbol{\omega}_1 \times \\ -\mathbf{d}_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} \boldsymbol{\omega}_1 \times \\ \mathbf{F}_{/\dot{\mathbf{d}}} \boldsymbol{\omega}_1 \times \\ \mathbf{d}_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} \boldsymbol{\omega}_1 \times \end{array} \right] \delta \mathbf{x}_1 + \left[\begin{array}{c} -\mathbf{F}_{/\dot{\mathbf{d}}} (\dot{\mathbf{d}}_1 \times -\boldsymbol{\omega}_1 \times \mathbf{d}_1 \times) \\ -\mathbf{d}_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} (\dot{\mathbf{d}}_1 \times -\boldsymbol{\omega}_1 \times \mathbf{d}_1 \times) \\ \mathbf{F}_{/\dot{\mathbf{d}}} (\dot{\mathbf{d}}_1 \times -\boldsymbol{\omega}_1 \times \mathbf{d}_1 \times) \\ \mathbf{d}_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} (\dot{\mathbf{d}}_1 \times -\boldsymbol{\omega}_1 \times \mathbf{d}_1 \times) \end{array} \right] \delta \mathbf{g}_1 \\ & + \left[\begin{array}{c} \mathbf{F}_{/\dot{\mathbf{d}}} \boldsymbol{\omega}_1 \times \\ \mathbf{d}_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} \boldsymbol{\omega}_1 \times \\ -\mathbf{F}_{/\dot{\mathbf{d}}} \boldsymbol{\omega}_1 \times \\ -\mathbf{d}_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} \boldsymbol{\omega}_1 \times \end{array} \right] \delta \mathbf{x}_2 + \left[\begin{array}{c} \mathbf{F}_{/\dot{\mathbf{d}}} (\dot{\mathbf{d}}_2 \times -\boldsymbol{\omega}_1 \times \mathbf{d}_2 \times) \\ \mathbf{d}_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} (\dot{\mathbf{d}}_2 \times -\boldsymbol{\omega}_1 \times \mathbf{d}_2 \times) \\ -\mathbf{F}_{/\dot{\mathbf{d}}} (\dot{\mathbf{d}}_2 \times -\boldsymbol{\omega}_1 \times \mathbf{d}_2 \times) \\ -\mathbf{d}_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} (\dot{\mathbf{d}}_2 \times -\boldsymbol{\omega}_1 \times \mathbf{d}_2 \times) \end{array} \right] \delta \mathbf{g}_2 \\ & + \left[\begin{array}{ccc} \mathbf{F}_{/\tilde{\mathbf{d}}} & -\mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{d}_1 \times & -\mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{d}_2 \times \\ \mathbf{d}_1 \times \mathbf{F}_{/\tilde{\mathbf{d}}} & -\mathbf{d}_1 \times \mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{d}_1 \times & -\mathbf{d}_1 \times \mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{d}_2 \times \\ -\mathbf{F}_{/\tilde{\mathbf{d}}} & \mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{d}_1 \times & \mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{d}_2 \times \\ -\mathbf{d}_2 \times \mathbf{F}_{/\tilde{\mathbf{d}}} & \mathbf{d}_2 \times \mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{d}_1 \times & \mathbf{d}_2 \times \mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{d}_2 \times \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_1 \\ \delta \mathbf{g}_1 \\ \delta \mathbf{x}_2 \\ \delta \mathbf{g}_2 \end{array} \right\} \\ & + \left[\begin{array}{cccc} \mathbf{0} & \mathbf{F} \times & \mathbf{0} & \mathbf{0} \\ -\mathbf{F} \times & \mathbf{d}_1 \times \mathbf{F} \times & \mathbf{F} \times & -\mathbf{F} \times \mathbf{d}_2 \times \\ \mathbf{0} & -\mathbf{F} \times & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{d}_2 \times \mathbf{F} \times & \mathbf{0} & \mathbf{F} \times \mathbf{d}_2 \times \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_1 \\ \delta \mathbf{g}_1 \\ \delta \mathbf{x}_2 \\ \delta \mathbf{g}_2 \end{array} \right\} = \left\{ \begin{array}{l} \mathbf{F} \\ \mathbf{d}_1 \times \mathbf{F} \\ -\mathbf{F} \\ -\mathbf{d}_2 \times \mathbf{F} \end{array} \right\} \end{aligned} \quad (9.305)$$

where

$$\mathbf{F}_{/\dot{\mathbf{d}}} = \mathbf{R}_{1h} \tilde{\mathbf{F}}_{/\dot{\mathbf{d}}} \mathbf{R}_{1h}^T \quad (9.306a)$$

$$\mathbf{F}_{/\tilde{\mathbf{d}}} = \mathbf{R}_{1h} \tilde{\mathbf{F}}_{/\tilde{\mathbf{d}}} \mathbf{R}_{1h}^T \quad (9.306b)$$

Invariant Deformable Displacement Joint

The invariant form of the **deformable displacement** joint assumes that the constitutive properties of the component, that explicitly depend only on the relative position of the nodes, is defined in a reference frame that is intermediate between those of node 1 and 2, much like the invariant form of the **deformable hinge** joint.

The distance between the two reference points, expressed in the global reference frame, is pulled back in the intermediate material reference frame by the transpose of matrix $\hat{\mathbf{R}}$

$$\tilde{\mathbf{d}} = \hat{\mathbf{R}}^T \mathbf{d} \quad (9.307)$$

The relative velocity is

$$\begin{aligned} \dot{\tilde{\mathbf{d}}} &= \dot{\hat{\mathbf{R}}}^T \mathbf{d} + \hat{\mathbf{R}}^T \dot{\mathbf{d}} \\ &= \hat{\mathbf{R}}^T (\dot{\mathbf{d}} - \hat{\boldsymbol{\omega}} \times \mathbf{d}) ; \end{aligned} \quad (9.308)$$

for a discussion of the properties of the entities that appear in Eq. (9.276) see Section 9.2.2.

The perturbation of the relative position yields

$$\delta\tilde{\mathbf{d}} = \hat{\mathbf{R}}^T (\delta\mathbf{d} + \mathbf{d} \times \hat{\boldsymbol{\theta}}_\delta) \quad (9.309)$$

where

$$\begin{aligned} \delta\mathbf{d} &= \delta\mathbf{x}_2 - \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} - \delta\mathbf{x}_1 + \mathbf{f}_1 \times \boldsymbol{\theta}_{1\delta} \\ &\stackrel{\text{uu}}{=} \delta\mathbf{x}_2 - \mathbf{f}_2 \times \delta\mathbf{g}_2 - \delta\mathbf{x}_1 + \mathbf{f}_1 \times \delta\mathbf{g}_1 \end{aligned} \quad (9.310a)$$

$$\begin{aligned} \hat{\boldsymbol{\theta}}_\delta &= \hat{\mathbf{I}}\boldsymbol{\theta}_{2\delta} + \hat{\mathbf{I}}^T\boldsymbol{\theta}_{1\delta} \\ &\stackrel{\text{uu}}{=} \hat{\mathbf{I}}\delta\mathbf{g}_2 + \hat{\mathbf{I}}^T\delta\mathbf{g}_1, \end{aligned} \quad (9.310b)$$

and thus

$$\begin{aligned} \delta\tilde{\mathbf{d}} &= \hat{\mathbf{R}}^T \left(\delta\mathbf{x}_2 + (\mathbf{d} \times \hat{\mathbf{I}} - \mathbf{f}_2 \times) \boldsymbol{\theta}_{2\delta} - \delta\mathbf{x}_1 + (\mathbf{d} \times \hat{\mathbf{I}}^T + \mathbf{f}_1 \times) \boldsymbol{\theta}_{1\delta} \right) \\ &\stackrel{\text{uu}}{=} \hat{\mathbf{R}}^T \left(\delta\mathbf{x}_2 + (\mathbf{d} \times \hat{\mathbf{I}} - \mathbf{f}_2 \times) \delta\mathbf{g}_2 - \delta\mathbf{x}_1 + (\mathbf{d} \times \hat{\mathbf{I}}^T + \mathbf{f}_1 \times) \delta\mathbf{g}_1 \right). \end{aligned} \quad (9.311)$$

The perturbation of the relative velocity yields

$$\delta\dot{\tilde{\mathbf{d}}} = \hat{\mathbf{R}}^T \left(\delta\dot{\mathbf{d}} + \mathbf{d} \times \delta\hat{\boldsymbol{\omega}} - \hat{\boldsymbol{\omega}} \times \delta\mathbf{d} + (\dot{\mathbf{d}} - \hat{\boldsymbol{\omega}} \times \mathbf{d}) \times \hat{\boldsymbol{\theta}}_\delta \right) \quad (9.312)$$

where

$$\begin{aligned} \delta\dot{\mathbf{d}} &= \delta\dot{\mathbf{x}}_2 - \mathbf{f}_2 \times \delta\boldsymbol{\omega}_2 - \boldsymbol{\omega}_2 \times \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} - \delta\dot{\mathbf{x}}_1 + \mathbf{f}_1 \times \delta\boldsymbol{\omega}_1 + \boldsymbol{\omega}_1 \times \mathbf{f}_1 \times \boldsymbol{\theta}_{1\delta} \\ &\stackrel{\text{uu}}{=} \delta\dot{\mathbf{x}}_2 - \mathbf{f}_2 \delta\mathbf{g}_2 + (\mathbf{f}_2 \times \boldsymbol{\omega}_2) \times \delta\mathbf{g}_2 - \delta\dot{\mathbf{x}}_1 + \mathbf{f}_1 \delta\mathbf{g}_1 - (\mathbf{f}_1 \times \boldsymbol{\omega}_1) \times \delta\mathbf{g}_1. \end{aligned} \quad (9.313)$$

The computation of $\delta\hat{\omega}$ has been illustrated in Section 9.2.2. Thus,

$$\begin{aligned}
\dot{\tilde{\mathbf{d}}} &= \hat{\mathbf{R}}^T \left(\delta\dot{\mathbf{x}}_2 - \mathbf{f}_2 \times \delta\omega_2 - \omega_2 \times \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} - \delta\dot{\mathbf{x}}_1 + \mathbf{f}_1 \times \delta\omega_1 + \omega_1 \times \mathbf{f}_1 \times \boldsymbol{\theta}_{1\delta} \right. \\
&\quad \left. + \mathbf{d} \times (\hat{\mathbf{I}}\delta\omega_2 + \hat{\mathbf{I}}^T\delta\omega_1 + \hat{\mathbf{I}}_2\boldsymbol{\theta}_{2\delta} + \hat{\mathbf{I}}_1\boldsymbol{\theta}_{1\delta}) \right. \\
&\quad \left. - \hat{\omega} \times (\delta\mathbf{x}_2 - \mathbf{f}_2 \times \boldsymbol{\theta}_{2\delta} - \delta\mathbf{x}_1 + \mathbf{f}_1 \times \boldsymbol{\theta}_{1\delta}) \right. \\
&\quad \left. + (\dot{\mathbf{d}} - \hat{\omega} \times \mathbf{d}) \times (\hat{\mathbf{I}}\boldsymbol{\theta}_{2\delta} + \hat{\mathbf{I}}^T\boldsymbol{\theta}_{1\delta}) \right) \\
&= \hat{\mathbf{R}}^T \left(\delta\dot{\mathbf{x}}_2 - \delta\dot{\mathbf{x}}_1 + (\mathbf{d} \times \hat{\mathbf{I}} - \mathbf{f}_2 \times) \delta\omega_2 + (\mathbf{d} \times \hat{\mathbf{I}}^T + \mathbf{f}_1 \times) \delta\omega_1 \right. \\
&\quad \left. - \hat{\omega} \times \delta\mathbf{x}_2 + \hat{\omega} \times \delta\mathbf{x}_1 \right. \\
&\quad \left. + ((\hat{\omega} - \omega_2) \times \mathbf{f}_2 \times + \mathbf{d} \times \hat{\mathbf{I}}_2 + (\dot{\mathbf{d}} - \hat{\omega} \times \mathbf{d}) \times \hat{\mathbf{I}}) \boldsymbol{\theta}_{2\delta} \right. \\
&\quad \left. + (-(\hat{\omega} - \omega_1) \times \mathbf{f}_1 \times + \mathbf{d} \times \hat{\mathbf{I}}_1 + (\dot{\mathbf{d}} - \hat{\omega} \times \mathbf{d}) \times \hat{\mathbf{I}}^T) \boldsymbol{\theta}_{1\delta} \right) \\
&\stackrel{\text{uu}}{=} \hat{\mathbf{R}}^T \left(\delta\dot{\mathbf{x}}_2 - \delta\dot{\mathbf{x}}_1 + (\mathbf{d} \times \hat{\mathbf{I}} - \mathbf{f}_2 \times) \delta\dot{\mathbf{g}}_2 + (\mathbf{d} \times \hat{\mathbf{I}}^T + \mathbf{f}_1 \times) \delta\dot{\mathbf{g}}_1 \right. \\
&\quad \left. - \hat{\omega} \times \delta\mathbf{x}_2 + \hat{\omega} \times \delta\mathbf{x}_1 \right. \\
&\quad \left. + \left(\begin{array}{l} (\hat{\mathbf{I}}^T(\omega_1 - \omega_2)) \times \mathbf{f}_2 \times - (\mathbf{d} \times \hat{\mathbf{I}} - \mathbf{f}_2 \times) \omega_2 \times \\ + \mathbf{d} \times \hat{\mathbf{I}}_2 + (\dot{\mathbf{d}} - \hat{\omega} \times \mathbf{d}) \times \hat{\mathbf{I}} \end{array} \right) \delta\mathbf{g}_2 \right. \\
&\quad \left. + \left(\begin{array}{l} (\hat{\mathbf{I}}(\omega_1 - \omega_2)) \times \mathbf{f}_1 \times - (\mathbf{d} \times \hat{\mathbf{I}}^T + \mathbf{f}_1 \times) \omega_1 \times \\ + \mathbf{d} \times \hat{\mathbf{I}}_1 + (\dot{\mathbf{d}} - \hat{\omega} \times \mathbf{d}) \times \hat{\mathbf{I}}^T \end{array} \right) \delta\mathbf{g}_1 \right) \tag{9.314}
\end{aligned}$$

The force and the moment result from the VWP

$$\begin{aligned}
\delta\mathcal{L} &= \delta\tilde{\mathbf{d}}^T \tilde{\mathbf{F}} \\
&= (\delta\mathbf{d} + \mathbf{d} \times \hat{\boldsymbol{\theta}}_\delta)^T \hat{\mathbf{R}} \tilde{\mathbf{F}} \\
&= (\delta\mathbf{x}_2 + (\mathbf{d} \times \hat{\mathbf{I}} - \mathbf{f}_2 \times) \boldsymbol{\theta}_{2\delta} - \delta\mathbf{x}_1 + (\mathbf{d} \times \hat{\mathbf{I}}^T + \mathbf{f}_1 \times) \boldsymbol{\theta}_{1\delta})^T \hat{\mathbf{R}} \tilde{\mathbf{F}} \tag{9.315}
\end{aligned}$$

and, after defining

$$\mathbf{F} = \hat{\mathbf{R}} \tilde{\mathbf{F}} (\tilde{\mathbf{d}}, \dot{\tilde{\mathbf{d}}}), \tag{9.316}$$

are

$$\mathbf{F}_1 = -\mathbf{F} \tag{9.317a}$$

$$\mathbf{M}_1 = -(\mathbf{f}_1 \times + \hat{\mathbf{I}}\mathbf{d} \times) \mathbf{F} \tag{9.317b}$$

$$\mathbf{F}_2 = \mathbf{F} \tag{9.317c}$$

$$\mathbf{M}_2 = (\mathbf{f}_2 \times - \hat{\mathbf{I}}^T\mathbf{d} \times) \mathbf{F}. \tag{9.317d}$$

Their linearization, after defining

$$\delta\mathbf{F} = \hat{\mathbf{R}} (\tilde{\mathbf{F}}_{/\tilde{\mathbf{d}}} \delta\tilde{\mathbf{d}} + \tilde{\mathbf{F}}_{/\dot{\tilde{\mathbf{d}}}} \delta\dot{\tilde{\mathbf{d}}}) - \mathbf{F} \times \hat{\boldsymbol{\theta}}_\delta, \tag{9.318}$$

yields

$$\delta \mathbf{F}_1 = -\delta \mathbf{F} \quad (9.319a)$$

$$\delta \mathbf{M}_1 = (\mathbf{f}_1 \times + \hat{\mathbf{I}} \mathbf{d} \times) \delta \mathbf{F} - \delta \mathbf{f}_1 \times \mathbf{F} - \delta \hat{\mathbf{I}} \mathbf{d} \times \mathbf{F} - \hat{\mathbf{I}} \delta \mathbf{d} \times \mathbf{F} \quad (9.319b)$$

$$\delta \mathbf{F}_2 = \delta \mathbf{F} \quad (9.319c)$$

$$\delta \mathbf{M}_2 = (\mathbf{f}_2 \times - \hat{\mathbf{I}}^T \mathbf{d} \times) \delta \mathbf{F} + \delta \mathbf{f}_2 \times \mathbf{F} - \delta \hat{\mathbf{I}}^T \mathbf{d} \times \mathbf{F} - \hat{\mathbf{I}}^T \delta \mathbf{d} \times \mathbf{F} \quad (9.319d)$$

After setting

$$\mathbf{F}_{/\ddot{\mathbf{d}}} = \hat{\mathbf{R}} \tilde{\mathbf{F}}_{/\ddot{\mathbf{d}}} \hat{\mathbf{R}}^T \quad (9.320a)$$

$$\mathbf{F}_{/\dot{\mathbf{d}}} = \hat{\mathbf{R}} \tilde{\mathbf{F}}_{/\dot{\mathbf{d}}} \hat{\mathbf{R}}^T \quad (9.320b)$$

and

$$\mathbf{A}_1 = (\mathbf{f}_1 \times + \hat{\mathbf{I}} \mathbf{d} \times) \quad (9.321a)$$

$$\mathbf{A}_2 = (\mathbf{f}_2 \times - \hat{\mathbf{I}}^T \mathbf{d} \times) \quad (9.321b)$$

$$\mathbf{B}_1 = ((\hat{\mathbf{I}}(\omega_1 - \omega_2)) \times \mathbf{f}_1 \times + \mathbf{d} \times \hat{\mathbf{I}}_1 + (\dot{\mathbf{d}} - \hat{\boldsymbol{\omega}} \times \mathbf{d}) \times \hat{\mathbf{I}}^T) \quad (9.321c)$$

$$\mathbf{B}_2 = ((\hat{\mathbf{I}}^T(\omega_1 - \omega_2)) \times \mathbf{f}_2 \times + \mathbf{d} \times \hat{\mathbf{I}}_2 + (\dot{\mathbf{d}} - \hat{\boldsymbol{\omega}} \times \mathbf{d}) \times \hat{\mathbf{I}}) \quad (9.321d)$$

the complete linearized problem is

$$\begin{aligned} & \left[\begin{array}{cccc} \mathbf{F}_{/\dot{\mathbf{d}}} & \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_1^T & -\mathbf{F}_{/\dot{\mathbf{d}}} & -\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_2^T \\ \mathbf{A}_1 \mathbf{F}_{/\dot{\mathbf{d}}} & \mathbf{A}_1 \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_1^T & -\mathbf{A}_1 \mathbf{F}_{/\dot{\mathbf{d}}} & -\mathbf{A}_1 \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_2^T \\ -\mathbf{F}_{/\dot{\mathbf{d}}} & -\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_1^T & \mathbf{F}_{/\dot{\mathbf{d}}} & \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_2^T \\ -\mathbf{A}_2 \mathbf{F}_{/\dot{\mathbf{d}}} & -\mathbf{A}_2 \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_1^T & \mathbf{A}_2 \mathbf{F}_{/\dot{\mathbf{d}}} & \mathbf{A}_2 \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_2^T \end{array} \right] \left\{ \begin{array}{l} \delta \dot{\mathbf{x}}_1 \\ \delta \boldsymbol{\omega}_1 \\ \delta \dot{\mathbf{x}}_2 \\ \delta \boldsymbol{\omega}_2 \end{array} \right\} \\ & + \left[\begin{array}{cccc} -\mathbf{F}_{/\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & -\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_1 & \mathbf{F}_{/\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & -\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_2 \\ -\mathbf{A}_1 \mathbf{F}_{/\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & -\mathbf{A}_1 \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_1 & \mathbf{A}_1 \mathbf{F}_{/\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & -\mathbf{A}_1 \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_2 \\ \mathbf{F}_{/\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_1 & -\mathbf{F}_{/\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_2 \\ \mathbf{A}_2 \mathbf{F}_{/\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & \mathbf{A}_2 \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_1 & -\mathbf{A}_2 \mathbf{F}_{/\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & \mathbf{A}_2 \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_2 \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_1 \\ \boldsymbol{\theta}_{1\delta} \\ \delta \mathbf{x}_2 \\ \boldsymbol{\theta}_{2\delta} \end{array} \right\} \\ & + \left[\begin{array}{cccc} \mathbf{F}_{/\ddot{\mathbf{d}}} & \mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{A}_1^T & -\mathbf{F}_{/\ddot{\mathbf{d}}} & -\mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{A}_2^T \\ \mathbf{A}_1 \mathbf{F}_{/\ddot{\mathbf{d}}} & \mathbf{A}_1 \mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{A}_1^T & -\mathbf{A}_1 \mathbf{F}_{/\ddot{\mathbf{d}}} & -\mathbf{A}_1 \mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{A}_2^T \\ -\mathbf{F}_{/\ddot{\mathbf{d}}} & -\mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{A}_1^T & \mathbf{F}_{/\ddot{\mathbf{d}}} & \mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{A}_2^T \\ -\mathbf{A}_2 \mathbf{F}_{/\ddot{\mathbf{d}}} & -\mathbf{A}_2 \mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{A}_1^T & \mathbf{A}_2 \mathbf{F}_{/\ddot{\mathbf{d}}} & \mathbf{A}_2 \mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{A}_2^T \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_1 \\ \boldsymbol{\theta}_{1\delta} \\ \delta \mathbf{x}_2 \\ \boldsymbol{\theta}_{2\delta} \end{array} \right\} \\ & + \left[\begin{array}{cccc} \mathbf{0} & \mathbf{F} \times \hat{\mathbf{I}}^T & & \\ -\hat{\mathbf{I}} \mathbf{F} \times & \mathbf{A}_1 \mathbf{F} \times \hat{\mathbf{I}}^T - \hat{\mathbf{I}}^T \mathbf{F} \times \mathbf{f}_1 \times - \hat{\mathbf{I}}_1 (\mathbf{d} \times \mathbf{F}) & & \\ \mathbf{0} & -\mathbf{F} \times \hat{\mathbf{I}}^T & & \\ -\hat{\mathbf{I}}^T \mathbf{F} \times & -\mathbf{A}_2 \mathbf{F} \times \hat{\mathbf{I}}^T - \hat{\mathbf{I}}^T \mathbf{F} \times \mathbf{f}_1 \times - (\hat{\mathbf{I}}^T)_{1(\mathbf{d} \times \mathbf{F})} & & \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_1 \\ \boldsymbol{\theta}_{1\delta} \end{array} \right\} \\ & + \left[\begin{array}{cccc} \mathbf{0} & \mathbf{F} \times \hat{\mathbf{I}} & & \\ \hat{\mathbf{I}} \mathbf{F} \times & \mathbf{A}_1 \mathbf{F} \times \hat{\mathbf{I}} - \hat{\mathbf{I}} \mathbf{F} \times \mathbf{f}_2 \times - \hat{\mathbf{I}}_2 (\mathbf{d} \times \mathbf{F}) & & \\ \mathbf{0} & -\mathbf{F} \times \hat{\mathbf{I}} & & \\ \hat{\mathbf{I}}^T \mathbf{F} \times & -\mathbf{A}_2 \mathbf{F} \times \hat{\mathbf{I}} + \hat{\mathbf{I}} \mathbf{F} \times \mathbf{f}_2 \times + (\hat{\mathbf{I}}^T)_{2(\mathbf{d} \times \mathbf{F})} & & \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_2 \\ \boldsymbol{\theta}_{2\delta} \end{array} \right\} \\ & = \left\{ \begin{array}{l} \mathbf{F} \\ \mathbf{A}_1 \mathbf{F} \\ -\mathbf{F} \\ -\mathbf{A}_2 \mathbf{F} \end{array} \right\} \end{aligned} \quad (9.322)$$

When the updated-updated approach is considered, after defining

$$\hat{\mathbf{B}}_1 = \mathbf{B}_1 + \mathbf{A}_1^T \boldsymbol{\omega}_1 \times \quad (9.323a)$$

$$\hat{\mathbf{B}}_2 = \mathbf{B}_2 - \mathbf{A}_2^T \boldsymbol{\omega}_2 \times, \quad (9.323b)$$

one obtains

$$\begin{aligned}
& \left[\begin{array}{cccc} \mathbf{F}/\dot{\mathbf{d}} & \mathbf{F}/\dot{\mathbf{d}} \mathbf{A}_1^T & -\mathbf{F}/\dot{\mathbf{d}} & -\mathbf{F}/\dot{\mathbf{d}} \mathbf{A}_2^T \\ \mathbf{A}_1 \mathbf{F}/\dot{\mathbf{d}} & \mathbf{A}_1 \mathbf{F}/\dot{\mathbf{d}} \mathbf{A}_1^T & -\mathbf{A}_1 \mathbf{F}/\dot{\mathbf{d}} & -\mathbf{A}_1 \mathbf{F}/\dot{\mathbf{d}} \mathbf{A}_2^T \\ -\mathbf{F}/\dot{\mathbf{d}} & -\mathbf{F}/\dot{\mathbf{d}} \mathbf{A}_1^T & \mathbf{F}/\dot{\mathbf{d}} & \mathbf{F}/\dot{\mathbf{d}} \mathbf{A}_2^T \\ -\mathbf{A}_2 \mathbf{F}/\dot{\mathbf{d}} & -\mathbf{A}_2 \mathbf{F}/\dot{\mathbf{d}} \mathbf{A}_1^T & \mathbf{A}_2 \mathbf{F}/\dot{\mathbf{d}} & \mathbf{A}_2 \mathbf{F}/\dot{\mathbf{d}} \mathbf{A}_2^T \end{array} \right] \left\{ \begin{array}{l} \delta \dot{\mathbf{x}}_1 \\ \delta \dot{\mathbf{g}}_1 \\ \delta \dot{\mathbf{x}}_2 \\ \delta \dot{\mathbf{g}}_2 \end{array} \right\} \\
& + \left[\begin{array}{cccc} -\mathbf{F}/\dot{\mathbf{d}} \hat{\boldsymbol{\omega}} \times & -\mathbf{F}/\dot{\mathbf{d}} \hat{\mathbf{B}}_1 & \mathbf{F}/\dot{\mathbf{d}} \hat{\boldsymbol{\omega}} \times & -\mathbf{F}/\dot{\mathbf{d}} \hat{\mathbf{B}}_2 \\ -\mathbf{A}_1 \mathbf{F}/\dot{\mathbf{d}} \hat{\boldsymbol{\omega}} \times & -\mathbf{A}_1 \mathbf{F}/\dot{\mathbf{d}} \hat{\mathbf{B}}_1 & \mathbf{A}_1 \mathbf{F}/\dot{\mathbf{d}} \hat{\boldsymbol{\omega}} \times & -\mathbf{A}_1 \mathbf{F}/\dot{\mathbf{d}} \hat{\mathbf{B}}_2 \\ \mathbf{F}/\dot{\mathbf{d}} \hat{\boldsymbol{\omega}} \times & \mathbf{F}/\dot{\mathbf{d}} \hat{\mathbf{B}}_1 & -\mathbf{F}/\dot{\mathbf{d}} \hat{\boldsymbol{\omega}} \times & \mathbf{F}/\dot{\mathbf{d}} \hat{\mathbf{B}}_2 \\ \mathbf{A}_2 \mathbf{F}/\dot{\mathbf{d}} \hat{\boldsymbol{\omega}} \times & \mathbf{A}_2 \mathbf{F}/\dot{\mathbf{d}} \hat{\mathbf{B}}_1 & -\mathbf{A}_2 \mathbf{F}/\dot{\mathbf{d}} \hat{\boldsymbol{\omega}} \times & \mathbf{A}_2 \mathbf{F}/\dot{\mathbf{d}} \hat{\mathbf{B}}_2 \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_1 \\ \delta \mathbf{g}_1 \\ \delta \mathbf{x}_2 \\ \delta \mathbf{g}_2 \end{array} \right\} \\
& + \left[\begin{array}{cccc} \mathbf{F}/\tilde{\mathbf{d}} & \mathbf{F}/\tilde{\mathbf{d}} \mathbf{A}_1^T & -\mathbf{F}/\tilde{\mathbf{d}} & -\mathbf{F}/\tilde{\mathbf{d}} \mathbf{A}_2^T \\ \mathbf{A}_1 \mathbf{F}/\tilde{\mathbf{d}} & \mathbf{A}_1 \mathbf{F}/\tilde{\mathbf{d}} \mathbf{A}_1^T & -\mathbf{A}_1 \mathbf{F}/\tilde{\mathbf{d}} & -\mathbf{A}_1 \mathbf{F}/\tilde{\mathbf{d}} \mathbf{A}_2^T \\ -\mathbf{F}/\tilde{\mathbf{d}} & -\mathbf{F}/\tilde{\mathbf{d}} \mathbf{A}_1^T & \mathbf{F}/\tilde{\mathbf{d}} & \mathbf{F}/\tilde{\mathbf{d}} \mathbf{A}_2^T \\ -\mathbf{A}_2 \mathbf{F}/\tilde{\mathbf{d}} & -\mathbf{A}_2 \mathbf{F}/\tilde{\mathbf{d}} \mathbf{A}_1^T & \mathbf{A}_2 \mathbf{F}/\tilde{\mathbf{d}} & \mathbf{A}_2 \mathbf{F}/\tilde{\mathbf{d}} \mathbf{A}_2^T \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_1 \\ \delta \mathbf{g}_1 \\ \delta \mathbf{x}_2 \\ \delta \mathbf{g}_2 \end{array} \right\} \\
& + \left[\begin{array}{cccc} \mathbf{0} & \mathbf{F} \times \hat{\mathbf{I}}^T & \mathbf{0} & \mathbf{0} \\ -\hat{\mathbf{I}} \mathbf{F} \times & \mathbf{A}_1 \mathbf{F} \times \hat{\mathbf{I}}^T - \hat{\mathbf{I}}^T \mathbf{F} \times \mathbf{f}_1 \times & -\hat{\mathbf{I}}_{1(\mathbf{d} \times \mathbf{F})} & \mathbf{0} \\ \mathbf{0} & -\mathbf{F} \times \hat{\mathbf{I}}^T & \mathbf{0} & \mathbf{0} \\ -\hat{\mathbf{I}}^T \mathbf{F} \times & -\mathbf{A}_2 \mathbf{F} \times \hat{\mathbf{I}}^T - \hat{\mathbf{I}}^T \mathbf{F} \times \mathbf{f}_1 \times & -(\hat{\mathbf{I}}^T)_{1(\mathbf{d} \times \mathbf{F})} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_1 \\ \delta \mathbf{g}_1 \end{array} \right\} \\
& + \left[\begin{array}{cccc} \mathbf{0} & \mathbf{F} \times \hat{\mathbf{I}} & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{I}} \mathbf{F} \times & \mathbf{A}_1 \mathbf{F} \times \hat{\mathbf{I}} - \hat{\mathbf{I}} \mathbf{F} \times \mathbf{f}_2 \times & -\hat{\mathbf{I}}_{2(\mathbf{d} \times \mathbf{F})} & \mathbf{0} \\ \mathbf{0} & -\mathbf{F} \times \hat{\mathbf{I}} & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{I}}^T \mathbf{F} \times & -\mathbf{A}_2 \mathbf{F} \times \hat{\mathbf{I}} + \hat{\mathbf{I}} \mathbf{F} \times \mathbf{f}_2 \times & +(\hat{\mathbf{I}}^T)_{2(\mathbf{d} \times \mathbf{F})} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_2 \\ \delta \mathbf{g}_2 \end{array} \right\} \\
& \stackrel{\text{uu}}{=} \left\{ \begin{array}{l} \mathbf{F} \\ \mathbf{A}_1 \mathbf{F} \\ -\mathbf{F} \\ -\mathbf{A}_2 \mathbf{F} \end{array} \right\} \quad (9.324)
\end{aligned}$$

9.2.4 Deformable Joint

The deformable joint applies an internal force and an internal moment to two nodes at a specified point that may be offset from the nodes; the force and the moment may depend on the relative position and velocity at the point of application and on the relative rotation vector and angular velocity of the nodes through a 6D constitutive law.

Although it may appear as a combination of the Deformable Hinge and the Deformable Displacement Joint, and despite some commonality of code, it is a bit more general because the internal force and moment may depend on the relative displacement and orientation, i.e. the displacements and the orientations are coupled. However, if a simple linear diagonal constitutive law is used, the same result with a bit less overhead is obtained by using a combination of a deformable hinge and a deformable displacement joint.

The deformable joint equations can be computed by combining those of the deformable hinge (Section 9.2.2) and of the deformable displacement joint (Section 9.2.3), considering that both the force \mathbf{F} and the moment \mathbf{M} simultaneously depend on the relative displacement, $\tilde{\mathbf{d}}$, the relative orientation, $\boldsymbol{\theta}$, and their time derivatives, $\dot{\tilde{\mathbf{d}}}$ and $\boldsymbol{\omega}$.

Attached Deformable Joint

Internal force and moment:

$$\mathbf{F}_i = (-1)^i \mathbf{R}_{1h} \tilde{\mathbf{F}}(\tilde{\mathbf{d}}, \boldsymbol{\theta}, \dot{\tilde{\mathbf{d}}}, \boldsymbol{\omega}) \quad (9.325a)$$

$$\mathbf{M}_i = \mathbf{d}_i \times \mathbf{F}_i + (-1)^i \mathbf{R}_{1h} \tilde{\mathbf{M}}(\tilde{\mathbf{d}}, \boldsymbol{\theta}, \dot{\tilde{\mathbf{d}}}, \boldsymbol{\omega}). \quad (9.325b)$$

Their linearization, after defining

$$\mathbf{F}_{/\tilde{\mathbf{d}}} = \mathbf{R}_{1h} \tilde{\mathbf{F}}_{/\tilde{\mathbf{d}}} \mathbf{R}_{1h}^T \quad (9.326a)$$

$$\mathbf{F}_{/\dot{\tilde{\mathbf{d}}}} = \mathbf{R}_{1h} \tilde{\mathbf{F}}_{/\dot{\tilde{\mathbf{d}}}} \mathbf{R}_{1h}^T \quad (9.326b)$$

$$\mathbf{F}_{/\boldsymbol{\theta}} = \mathbf{R}_{1h} \tilde{\mathbf{F}}_{/\boldsymbol{\theta}} \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1h}^T \quad (9.326c)$$

$$\mathbf{F}_{/\boldsymbol{\omega}} = \mathbf{R}_{1h} \tilde{\mathbf{F}}_{/\boldsymbol{\omega}} \mathbf{R}_{1h}^T \quad (9.326d)$$

$$\mathbf{M}_{/\tilde{\mathbf{d}}} = \mathbf{R}_{1h} \tilde{\mathbf{M}}_{/\tilde{\mathbf{d}}} \mathbf{R}_{1h}^T \quad (9.326e)$$

$$\mathbf{M}_{/\dot{\tilde{\mathbf{d}}}} = \mathbf{R}_{1h} \tilde{\mathbf{M}}_{/\dot{\tilde{\mathbf{d}}}} \mathbf{R}_{1h}^T \quad (9.326f)$$

$$\mathbf{M}_{/\boldsymbol{\theta}} = \mathbf{R}_{1h} \tilde{\mathbf{M}}_{/\boldsymbol{\theta}} \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1h}^T \quad (9.326g)$$

$$\mathbf{M}_{/\boldsymbol{\omega}} = \mathbf{R}_{1h} \tilde{\mathbf{M}}_{/\boldsymbol{\omega}} \mathbf{R}_{1h}^T, \quad (9.326h)$$

yields the viscous contribution to perturbation,

$$\begin{aligned}
& \left[\begin{array}{c} \mathbf{F}_{/\dot{\mathbf{d}}} \\ d_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} + \mathbf{M}_{/\dot{\mathbf{d}}} \\ -\mathbf{F}_{/\dot{\mathbf{d}}} \\ -d_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} - \mathbf{M}_{/\dot{\mathbf{d}}} \end{array} \right] \delta \dot{\mathbf{x}}_1 + \left[\begin{array}{c} -\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_1 \times + \mathbf{F}_{/\omega} \\ -d_1 \times (\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_1 \times - \mathbf{F}_{/\omega}) - \mathbf{M}_{/\dot{\mathbf{d}}} \mathbf{d}_1 \times + \mathbf{M}_{/\omega} \\ \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_1 \times - \mathbf{F}_{/\omega} \\ d_2 \times (\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_1 \times - \mathbf{F}_{/\omega}) + \mathbf{M}_{/\dot{\mathbf{d}}} \mathbf{d}_1 \times - \mathbf{M}_{/\omega} \end{array} \right] \delta \dot{\mathbf{g}}_1 \\
& + \left[\begin{array}{c} -\mathbf{F}_{/\dot{\mathbf{d}}} \\ -d_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} - \mathbf{M}_{/\dot{\mathbf{d}}} \\ \mathbf{F}_{/\dot{\mathbf{d}}} \\ d_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} + \mathbf{M}_{/\dot{\mathbf{d}}} \end{array} \right] \delta \dot{\mathbf{x}}_2 + \left[\begin{array}{c} \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_2 \times - \mathbf{F}_{/\omega} \\ d_1 \times (\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_2 \times - \mathbf{F}_{/\omega}) + \mathbf{M}_{/\dot{\mathbf{d}}} \mathbf{d}_2 \times - \mathbf{M}_{/\omega} \\ -\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_2 \times + \mathbf{F}_{/\omega} \\ -d_2 \times (\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{d}_2 \times - \mathbf{F}_{/\omega}) - \mathbf{M}_{/\dot{\mathbf{d}}} \mathbf{d}_2 \times + \mathbf{M}_{/\omega} \end{array} \right] \delta \dot{\mathbf{g}}_2 \\
& \quad + \left[\begin{array}{c} -\mathbf{F}_{/\dot{\mathbf{d}}} \boldsymbol{\omega}_1 \times \\ -(\mathbf{d}_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} + \mathbf{M}_{/\dot{\mathbf{d}}}) \boldsymbol{\omega}_1 \times \\ \mathbf{F}_{/\dot{\mathbf{d}}} \boldsymbol{\omega}_1 \times \\ (\mathbf{d}_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} + \mathbf{M}_{/\dot{\mathbf{d}}}) \boldsymbol{\omega}_1 \times \end{array} \right] \delta \boldsymbol{x}_1 \\
& \quad + \left[\begin{array}{c} -\mathbf{F}_{/\dot{\mathbf{d}}} (\dot{\mathbf{d}}_1 \times - \boldsymbol{\omega}_1 \times \mathbf{d}_1 \times) - \mathbf{F}_{/\omega} \boldsymbol{\omega}_2 \times \\ -(\mathbf{d}_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} + \mathbf{M}_{/\dot{\mathbf{d}}}) (\dot{\mathbf{d}}_1 \times - \boldsymbol{\omega}_1 \times \mathbf{d}_1 \times) - (\mathbf{d}_1 \times \mathbf{F}_{/\omega} + \mathbf{M}_{/\omega}) \boldsymbol{\omega}_2 \times \\ \mathbf{F}_{/\dot{\mathbf{d}}} (\dot{\mathbf{d}}_1 \times - \boldsymbol{\omega}_1 \times \mathbf{d}_1 \times) + \mathbf{F}_{/\omega} \boldsymbol{\omega}_2 \times \\ (\mathbf{d}_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} + \mathbf{M}_{/\dot{\mathbf{d}}}) (\dot{\mathbf{d}}_1 \times - \boldsymbol{\omega}_1 \times \mathbf{d}_1 \times) + (\mathbf{d}_2 \times \mathbf{F}_{/\omega} + \mathbf{M}_{/\omega}) \boldsymbol{\omega}_2 \times \end{array} \right] \delta \boldsymbol{g}_1 \\
& \quad + \left[\begin{array}{c} \mathbf{F}_{/\dot{\mathbf{d}}} \boldsymbol{\omega}_1 \times \\ (\mathbf{d}_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} + \mathbf{M}_{/\dot{\mathbf{d}}}) \boldsymbol{\omega}_1 \times \\ -\mathbf{F}_{/\dot{\mathbf{d}}} \boldsymbol{\omega}_1 \times \\ -(\mathbf{d}_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} + \mathbf{M}_{/\dot{\mathbf{d}}}) \boldsymbol{\omega}_1 \times \end{array} \right] \delta \boldsymbol{x}_2 \\
& \quad + \left[\begin{array}{c} \mathbf{F}_{/\dot{\mathbf{d}}} (\dot{\mathbf{d}}_2 \times - \boldsymbol{\omega}_1 \times \mathbf{d}_2 \times) + \mathbf{F}_{/\omega} \boldsymbol{\omega}_2 \times \\ (\mathbf{d}_1 \times \mathbf{F}_{/\dot{\mathbf{d}}} + \mathbf{M}_{/\dot{\mathbf{d}}}) (\dot{\mathbf{d}}_2 \times - \boldsymbol{\omega}_1 \times \mathbf{d}_2 \times) + (\mathbf{d}_1 \times \mathbf{F}_{/\omega} + \mathbf{M}_{/\omega}) \boldsymbol{\omega}_2 \times \\ -\mathbf{F}_{/\dot{\mathbf{d}}} (\dot{\mathbf{d}}_2 \times - \boldsymbol{\omega}_1 \times \mathbf{d}_2 \times) - \mathbf{F}_{/\omega} \boldsymbol{\omega}_2 \times \\ -(\mathbf{d}_2 \times \mathbf{F}_{/\dot{\mathbf{d}}} + \mathbf{M}_{/\dot{\mathbf{d}}}) (\dot{\mathbf{d}}_2 \times - \boldsymbol{\omega}_1 \times \mathbf{d}_2 \times) + (\mathbf{d}_2 \times \mathbf{F}_{/\omega} + \mathbf{M}_{/\omega}) \boldsymbol{\omega}_2 \times \end{array} \right] \delta \boldsymbol{g}_2 \tag{9.327}
\end{aligned}$$

the elastic contribution to perturbation,

$$\begin{aligned}
& \left[\begin{array}{c} \mathbf{F}_{/\ddot{\mathbf{d}}} \\ d_1 \times \mathbf{F}_{/\ddot{\mathbf{d}}} + \mathbf{M}_{/\ddot{\mathbf{d}}} \\ -\mathbf{F}_{/\ddot{\mathbf{d}}} \\ -d_2 \times \mathbf{F}_{/\ddot{\mathbf{d}}} - \mathbf{M}_{/\ddot{\mathbf{d}}} \end{array} \right] \delta \boldsymbol{x}_1 + \left[\begin{array}{c} -\mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{d}_1 \times + \mathbf{F}_{/\theta} \\ -d_1 \times (\mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{d}_1 \times - \mathbf{F}_{/\theta}) - \mathbf{M}_{/\ddot{\mathbf{d}}} \mathbf{d}_1 \times + \mathbf{M}_{/\theta} \\ \mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{d}_1 \times - \mathbf{F}_{/\theta} \\ d_2 \times (\mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{d}_1 \times - \mathbf{F}_{/\theta}) + \mathbf{M}_{/\ddot{\mathbf{d}}} \mathbf{d}_1 \times - \mathbf{M}_{/\theta} \end{array} \right] \delta \boldsymbol{g}_1 \\
& + \left[\begin{array}{c} -\mathbf{F}_{/\ddot{\mathbf{d}}} \\ -d_1 \times \mathbf{F}_{/\ddot{\mathbf{d}}} - \mathbf{M}_{/\ddot{\mathbf{d}}} \\ \mathbf{F}_{/\ddot{\mathbf{d}}} \\ d_2 \times \mathbf{F}_{/\ddot{\mathbf{d}}} + \mathbf{M}_{/\ddot{\mathbf{d}}} \end{array} \right] \delta \boldsymbol{x}_2 + \left[\begin{array}{c} \mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{d}_2 \times - \mathbf{F}_{/\theta} \\ d_1 \times (\mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{d}_2 \times - \mathbf{F}_{/\theta}) + \mathbf{M}_{/\ddot{\mathbf{d}}} \mathbf{d}_2 \times - \mathbf{M}_{/\theta} \\ -\mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{d}_2 \times + \mathbf{F}_{/\theta} \\ -d_2 \times (\mathbf{F}_{/\ddot{\mathbf{d}}} \mathbf{d}_2 \times - \mathbf{F}_{/\theta}) - \mathbf{M}_{/\ddot{\mathbf{d}}} \mathbf{d}_2 \times + \mathbf{M}_{/\theta} \end{array} \right] \delta \boldsymbol{g}_2 \tag{9.328}
\end{aligned}$$

and a common contribution to perturbation,

$$\left[\begin{array}{cccc} 0 & \mathbf{F} \times & 0 & 0 \\ -\mathbf{F} \times & d_1 \times \mathbf{F} \times + \mathbf{M} \times & \mathbf{F} \times & -\mathbf{F} \times \mathbf{d}_2 \times \\ 0 & -\mathbf{F} \times & 0 & 0 \\ 0 & -d_2 \times \mathbf{F} \times - \mathbf{M} \times & 0 & \mathbf{F} \times \mathbf{d}_2 \times \end{array} \right] \left\{ \begin{array}{c} \delta \boldsymbol{x}_1 \\ \delta \boldsymbol{g}_1 \\ \delta \boldsymbol{x}_2 \\ \delta \boldsymbol{g}_2 \end{array} \right\} \stackrel{\text{uu}}{=} \left\{ \begin{array}{c} \mathbf{F} \\ d_1 \times \mathbf{F} + \mathbf{M} \\ -\mathbf{F} \\ -d_2 \times \mathbf{F} - \mathbf{M} \end{array} \right\} \tag{9.329}$$

The contribution of Eq. (9.329) is always present. The contribution of Eq. (9.328) is present in the elastic and in the viscoelastic variants of the joint. The contribution of Eq. (9.327) is present in the viscous and in the viscoelastic variants of the joint.

Invariant Deformable Joint

The force and the moment, after defining

$$\mathbf{F} = \hat{\mathbf{R}}\tilde{\mathbf{F}}(\tilde{\mathbf{d}}, \dot{\tilde{\mathbf{d}}}, \boldsymbol{\theta}, \bar{\boldsymbol{\omega}}) \quad (9.330a)$$

$$\mathbf{M} = \hat{\mathbf{R}}\tilde{\mathbf{M}}(\tilde{\mathbf{d}}, \dot{\tilde{\mathbf{d}}}, \boldsymbol{\theta}, \bar{\boldsymbol{\omega}}), \quad (9.330b)$$

are

$$\mathbf{F}_1 = -\mathbf{F} \quad (9.331a)$$

$$\mathbf{M}_1 = -(\mathbf{f}_1 \times + \hat{\mathbf{I}}\mathbf{d} \times) \mathbf{F} - \mathbf{M} \quad (9.331b)$$

$$\mathbf{F}_2 = \mathbf{F} \quad (9.331c)$$

$$\mathbf{M}_2 = (\mathbf{f}_2 \times - \hat{\mathbf{I}}^T \mathbf{d} \times) \mathbf{F} + \mathbf{M}. \quad (9.331d)$$

Their linearization, after defining

$$\mathbf{F}_{/\tilde{\mathbf{d}}} = \hat{\mathbf{R}}\tilde{\mathbf{F}}_{/\tilde{\mathbf{d}}}\hat{\mathbf{R}}^T \quad (9.332a)$$

$$\mathbf{F}_{/\dot{\tilde{\mathbf{d}}}} = \hat{\mathbf{R}}\tilde{\mathbf{F}}_{/\dot{\tilde{\mathbf{d}}}}\hat{\mathbf{R}}^T \quad (9.332b)$$

$$\mathbf{F}_{/\boldsymbol{\theta}} = \hat{\mathbf{R}}\tilde{\mathbf{F}}_{/\boldsymbol{\theta}}\boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1h}^T \quad (9.332c)$$

$$\mathbf{F}_{/\boldsymbol{\omega}} = \hat{\mathbf{R}}\tilde{\mathbf{F}}_{/\boldsymbol{\omega}}\hat{\mathbf{R}}^T \quad (9.332d)$$

$$\mathbf{M}_{/\tilde{\mathbf{d}}} = \hat{\mathbf{R}}\tilde{\mathbf{M}}_{/\tilde{\mathbf{d}}}\hat{\mathbf{R}}^T \quad (9.332e)$$

$$\mathbf{M}_{/\dot{\tilde{\mathbf{d}}}} = \hat{\mathbf{R}}\tilde{\mathbf{M}}_{/\dot{\tilde{\mathbf{d}}}}\hat{\mathbf{R}}^T \quad (9.332f)$$

$$\mathbf{M}_{/\boldsymbol{\theta}} = \hat{\mathbf{R}}\tilde{\mathbf{M}}_{/\boldsymbol{\theta}}\boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_{1h}^T \quad (9.332g)$$

$$\mathbf{M}_{/\boldsymbol{\omega}} = \hat{\mathbf{R}}\tilde{\mathbf{M}}_{/\boldsymbol{\omega}}\hat{\mathbf{R}}^T, \quad (9.332h)$$

and exploiting the definitions of Eqs (9.321) and

$$\mathbf{C}_1 = (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \hat{\mathbf{I}}^T \quad (9.333a)$$

$$\mathbf{C}_2 = (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \hat{\mathbf{I}}, \quad (9.333b)$$

yields

$$\begin{aligned}
& \left[\begin{array}{cc} \frac{\mathbf{F}}{\dot{\mathbf{d}}} & \frac{\mathbf{F}}{\dot{\mathbf{d}}} \mathbf{A}_1^T + \mathbf{F}/\omega \\ \mathbf{A}_1 \frac{\mathbf{F}}{\dot{\mathbf{d}}} & \mathbf{A}_1 (\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_1^T + \mathbf{F}/\omega) + \mathbf{M}_{/\dot{\mathbf{d}}} \mathbf{A}_1^T + \mathbf{M}/\omega \\ -\mathbf{F}_{/\dot{\mathbf{d}}} & -\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_1^T - \mathbf{F}/\omega \\ -\mathbf{A}_2 \frac{\mathbf{F}}{\dot{\mathbf{d}}} & -\mathbf{A}_2 (\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_1^T + \mathbf{F}/\omega) - \mathbf{M}_{/\dot{\mathbf{d}}} \mathbf{A}_1^T - \mathbf{M}/\omega \end{array} \right] \left\{ \begin{array}{c} \delta \dot{\mathbf{x}}_1 \\ \delta \boldsymbol{\omega}_1 \end{array} \right\} \\
& + \left[\begin{array}{cc} -\mathbf{F}_{/\dot{\mathbf{d}}} & -\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_2^T - \mathbf{F}/\omega \\ -\mathbf{A}_1 \frac{\mathbf{F}}{\dot{\mathbf{d}}} & -\mathbf{A}_1 (\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_2^T + \mathbf{F}/\omega) - \mathbf{M}_{/\dot{\mathbf{d}}} \mathbf{A}_2^T - \mathbf{M}/\omega \\ \mathbf{F}_{/\dot{\mathbf{d}}} & \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_2^T + \mathbf{F}/\omega \\ \mathbf{A}_2 \frac{\mathbf{F}}{\dot{\mathbf{d}}} & \mathbf{A}_2 (\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{A}_2^T + \mathbf{F}/\omega) + \mathbf{M}_{/\dot{\mathbf{d}}} \mathbf{A}_2^T + \mathbf{M}/\omega \end{array} \right] \left\{ \begin{array}{c} \delta \dot{\mathbf{x}}_2 \\ \delta \boldsymbol{\omega}_2 \end{array} \right\} \\
& + \left[\begin{array}{cc} -\mathbf{F}_{/\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & -\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_1 - \mathbf{F}/\omega \mathbf{C}_1 \\ -\mathbf{A}_1 \frac{\mathbf{F}}{\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & -\mathbf{A}_1 (\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_1 + \mathbf{F}/\omega \mathbf{C}_1) - \mathbf{M}_{/\dot{\mathbf{d}}} \mathbf{B}_1 - \mathbf{M}/\omega \mathbf{C}_1 \\ \mathbf{F}_{/\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_1 + \mathbf{F}/\omega \mathbf{C}_1 \\ \mathbf{A}_2 \frac{\mathbf{F}}{\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & \mathbf{A}_2 (\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_1 + \mathbf{F}/\omega \mathbf{C}_1) + \mathbf{M}_{/\dot{\mathbf{d}}} \mathbf{B}_1 + \mathbf{M}/\omega \mathbf{C}_1 \end{array} \right] \left\{ \begin{array}{c} \delta \mathbf{x}_1 \\ \boldsymbol{\theta}_{1\delta} \end{array} \right\} \\
& + \left[\begin{array}{cc} \mathbf{F}_{/\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & -\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_2 - \mathbf{F}/\omega \mathbf{C}_2 \\ \mathbf{A}_1 \frac{\mathbf{F}}{\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & -\mathbf{A}_1 (\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_2 + \mathbf{F}/\omega \mathbf{C}_2) - \mathbf{M}_{/\dot{\mathbf{d}}} \mathbf{B}_2 - \mathbf{M}/\omega \mathbf{C}_2 \\ -\mathbf{F}_{/\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & \mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_2 + \mathbf{F}/\omega \mathbf{C}_2 \\ -\mathbf{A}_2 \frac{\mathbf{F}}{\dot{\mathbf{d}}} \hat{\boldsymbol{\omega}} \times & \mathbf{A}_2 (\mathbf{F}_{/\dot{\mathbf{d}}} \mathbf{B}_2 + \mathbf{F}/\omega \mathbf{C}_2) + \mathbf{M}_{/\dot{\mathbf{d}}} \mathbf{B}_2 + \mathbf{M}/\omega \mathbf{C}_2 \end{array} \right] \left\{ \begin{array}{c} \delta \mathbf{x}_2 \\ \boldsymbol{\theta}_{2\delta} \end{array} \right\} \\
& + \left[\begin{array}{cc} \mathbf{F}_{/\tilde{\mathbf{d}}} & \mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{A}_1^T + \mathbf{F}/\boldsymbol{\theta} \\ \mathbf{A}_1 \frac{\mathbf{F}}{\tilde{\mathbf{d}}} & \mathbf{A}_1 (\mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{A}_1^T + \mathbf{F}/\boldsymbol{\theta}) + \mathbf{M}_{/\tilde{\mathbf{d}}} \mathbf{A}_1^T + \mathbf{M}/\boldsymbol{\theta} \\ -\mathbf{F}_{/\tilde{\mathbf{d}}} & -\mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{A}_1^T - \mathbf{F}/\boldsymbol{\theta} \\ -\mathbf{A}_2 \frac{\mathbf{F}}{\tilde{\mathbf{d}}} & -\mathbf{A}_2 (\mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{A}_1^T + \mathbf{F}/\boldsymbol{\theta}) - \mathbf{M}_{/\tilde{\mathbf{d}}} \mathbf{A}_1^T - \mathbf{M}/\boldsymbol{\theta} \end{array} \right] \left\{ \begin{array}{c} \delta \mathbf{x}_1 \\ \boldsymbol{\theta}_{1\delta} \end{array} \right\} \\
& + \left[\begin{array}{cc} -\mathbf{F}_{/\tilde{\mathbf{d}}} & -\mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{A}_2^T - \mathbf{F}/\boldsymbol{\theta} \\ -\mathbf{A}_1 \frac{\mathbf{F}}{\tilde{\mathbf{d}}} & -\mathbf{A}_1 (\mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{A}_2^T + \mathbf{F}/\boldsymbol{\theta}) - \mathbf{M}_{/\tilde{\mathbf{d}}} \mathbf{A}_2^T - \mathbf{M}/\boldsymbol{\theta} \\ \mathbf{F}_{/\tilde{\mathbf{d}}} & \mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{A}_2^T + \mathbf{F}/\boldsymbol{\theta} \\ \mathbf{A}_2 \frac{\mathbf{F}}{\tilde{\mathbf{d}}} & \mathbf{A}_2 (\mathbf{F}_{/\tilde{\mathbf{d}}} \mathbf{A}_2^T + \mathbf{F}/\boldsymbol{\theta}) + \mathbf{M}_{/\tilde{\mathbf{d}}} \mathbf{A}_2^T + \mathbf{M}/\boldsymbol{\theta} \end{array} \right] \left\{ \begin{array}{c} \delta \mathbf{x}_2 \\ \boldsymbol{\theta}_{2\delta} \end{array} \right\} \\
& + \left[\begin{array}{cc} \mathbf{0} & \mathbf{F} \times \hat{\mathbf{I}}^T \\ -\hat{\mathbf{I}} \mathbf{F} \times & (\mathbf{A}_1 \mathbf{F} \times + \mathbf{M} \times) \hat{\mathbf{I}}^T - \hat{\mathbf{I}}^T \mathbf{F} \times \mathbf{f}_1 \times - \hat{\mathbf{I}}_{1(\mathbf{d} \times \mathbf{F})} \\ \mathbf{0} & -\mathbf{F} \times \hat{\mathbf{I}}^T \\ -\hat{\mathbf{I}}^T \mathbf{F} \times & -(\mathbf{A}_2 \mathbf{F} \times + \mathbf{M} \times) \hat{\mathbf{I}}^T - \hat{\mathbf{I}}^T \mathbf{F} \times \mathbf{f}_1 \times - (\hat{\mathbf{I}}^T)_{1(\mathbf{d} \times \mathbf{F})} \end{array} \right] \left\{ \begin{array}{c} \delta \mathbf{x}_1 \\ \boldsymbol{\theta}_{1\delta} \end{array} \right\} \\
& + \left[\begin{array}{cc} \mathbf{0} & \mathbf{F} \times \hat{\mathbf{I}} \\ \hat{\mathbf{I}} \mathbf{F} \times & (\mathbf{A}_1 \mathbf{F} \times + \mathbf{M} \times) \hat{\mathbf{I}} - \hat{\mathbf{I}} \mathbf{F} \times \mathbf{f}_2 \times - \hat{\mathbf{I}}_{2(\mathbf{d} \times \mathbf{F})} \\ \mathbf{0} & -\mathbf{F} \times \hat{\mathbf{I}} \\ \hat{\mathbf{I}}^T \mathbf{F} \times & -(\mathbf{A}_2 \mathbf{F} \times + \mathbf{M} \times) \hat{\mathbf{I}} + \hat{\mathbf{I}} \mathbf{F} \times \mathbf{f}_2 \times + (\hat{\mathbf{I}}^T)_{2(\mathbf{d} \times \mathbf{F})} \end{array} \right] \left\{ \begin{array}{c} \delta \mathbf{x}_2 \\ \boldsymbol{\theta}_{2\delta} \end{array} \right\} \\
& = \left\{ \begin{array}{c} \mathbf{F} \\ \mathbf{A}_1 \mathbf{F} + \mathbf{M} \\ -\mathbf{F} \\ -\mathbf{A}_2 \mathbf{F} - \mathbf{M} \end{array} \right\} \quad (9.334)
\end{aligned}$$

The updated-updated form is

$$\begin{aligned}
& \left[\begin{array}{cc} \mathbf{F}_{/\dot{\tilde{d}}} & \mathbf{F}_{/\dot{\tilde{d}}} \mathbf{A}_1^T + \mathbf{F}_{/\omega} \\ \mathbf{A}_1 \mathbf{F}_{/\dot{\tilde{d}}} & \mathbf{A}_1 (\mathbf{F}_{/\dot{\tilde{d}}} \mathbf{A}_1^T + \mathbf{F}_{/\omega}) + \mathbf{M}_{/\dot{\tilde{d}}} \mathbf{A}_1^T + \mathbf{M}_{/\omega} \\ -\mathbf{F}_{/\dot{\tilde{d}}} & -\mathbf{F}_{/\dot{\tilde{d}}} \mathbf{A}_1^T - \mathbf{F}_{/\omega} \\ -\mathbf{A}_2 \mathbf{F}_{/\dot{\tilde{d}}} & -\mathbf{A}_2 (\mathbf{F}_{/\dot{\tilde{d}}} \mathbf{A}_1^T + \mathbf{F}_{/\omega}) - \mathbf{M}_{/\dot{\tilde{d}}} \mathbf{A}_1^T - \mathbf{M}_{/\omega} \end{array} \right] \left\{ \begin{array}{l} \delta \dot{\mathbf{x}}_1 \\ \delta \dot{\mathbf{g}}_1 \end{array} \right\} \\
& + \left[\begin{array}{cc} -\mathbf{F}_{/\dot{\tilde{d}}} & -\mathbf{F}_{/\dot{\tilde{d}}} \mathbf{A}_2^T - \mathbf{F}_{/\omega} \\ -\mathbf{A}_1 \mathbf{F}_{/\dot{\tilde{d}}} & -\mathbf{A}_1 (\mathbf{F}_{/\dot{\tilde{d}}} \mathbf{A}_2^T + \mathbf{F}_{/\omega}) - \mathbf{M}_{/\dot{\tilde{d}}} \mathbf{A}_2^T - \mathbf{M}_{/\omega} \\ \mathbf{F}_{/\dot{\tilde{d}}} & \mathbf{F}_{/\dot{\tilde{d}}} \mathbf{A}_2^T + \mathbf{F}_{/\omega} \\ \mathbf{A}_2 \mathbf{F}_{/\dot{\tilde{d}}} & \mathbf{A}_2 (\mathbf{F}_{/\dot{\tilde{d}}} \mathbf{A}_2^T + \mathbf{F}_{/\omega}) + \mathbf{M}_{/\dot{\tilde{d}}} \mathbf{A}_2^T + \mathbf{M}_{/\omega} \end{array} \right] \left\{ \begin{array}{l} \delta \dot{\mathbf{x}}_2 \\ \delta \dot{\mathbf{g}}_2 \end{array} \right\} \\
& + \left[\begin{array}{cc} -\mathbf{F}_{/\dot{\tilde{d}}} \hat{\boldsymbol{\omega}} \times & -\mathbf{F}_{/\dot{\tilde{d}}} \hat{\mathbf{B}}_1 - \mathbf{F}_{/\omega} \hat{\mathbf{C}}_1 \\ -\mathbf{A}_1 \mathbf{F}_{/\dot{\tilde{d}}} \hat{\boldsymbol{\omega}} \times & -\mathbf{A}_1 (\mathbf{F}_{/\dot{\tilde{d}}} \hat{\mathbf{B}}_1 + \mathbf{F}_{/\omega} \hat{\mathbf{C}}_1) - \mathbf{M}_{/\dot{\tilde{d}}} \hat{\mathbf{B}}_1 - \mathbf{M}_{/\omega} \hat{\mathbf{C}}_1 \\ \mathbf{F}_{/\dot{\tilde{d}}} \hat{\boldsymbol{\omega}} \times & \mathbf{F}_{/\dot{\tilde{d}}} \hat{\mathbf{B}}_1 + \mathbf{F}_{/\omega} \hat{\mathbf{C}}_1 \\ \mathbf{A}_2 \mathbf{F}_{/\dot{\tilde{d}}} \hat{\boldsymbol{\omega}} \times & \mathbf{A}_2 (\mathbf{F}_{/\dot{\tilde{d}}} \hat{\mathbf{B}}_1 + \mathbf{F}_{/\omega} \hat{\mathbf{C}}_1) + \mathbf{M}_{/\dot{\tilde{d}}} \hat{\mathbf{B}}_1 + \mathbf{M}_{/\omega} \hat{\mathbf{C}}_1 \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_1 \\ \delta \mathbf{g}_1 \end{array} \right\} \\
& + \left[\begin{array}{cc} \mathbf{F}_{/\dot{\tilde{d}}} \hat{\boldsymbol{\omega}} \times & -\mathbf{F}_{/\dot{\tilde{d}}} \hat{\mathbf{B}}_2 - \mathbf{F}_{/\omega} \hat{\mathbf{C}}_2 \\ \mathbf{A}_1 \mathbf{F}_{/\dot{\tilde{d}}} \hat{\boldsymbol{\omega}} \times & -\mathbf{A}_1 (\mathbf{F}_{/\dot{\tilde{d}}} \hat{\mathbf{B}}_2 + \mathbf{F}_{/\omega} \hat{\mathbf{C}}_2) - \mathbf{M}_{/\dot{\tilde{d}}} \hat{\mathbf{B}}_2 - \mathbf{M}_{/\omega} \hat{\mathbf{C}}_2 \\ -\mathbf{F}_{/\dot{\tilde{d}}} \hat{\boldsymbol{\omega}} \times & \mathbf{F}_{/\dot{\tilde{d}}} \hat{\mathbf{B}}_2 + \mathbf{F}_{/\omega} \hat{\mathbf{C}}_2 \\ -\mathbf{A}_2 \mathbf{F}_{/\dot{\tilde{d}}} \hat{\boldsymbol{\omega}} \times & \mathbf{A}_2 (\mathbf{F}_{/\dot{\tilde{d}}} \hat{\mathbf{B}}_2 + \mathbf{F}_{/\omega} \hat{\mathbf{C}}_2) + \mathbf{M}_{/\dot{\tilde{d}}} \hat{\mathbf{B}}_2 + \mathbf{M}_{/\omega} \hat{\mathbf{C}}_2 \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_2 \\ \delta \mathbf{g}_2 \end{array} \right\} \\
& + \left[\begin{array}{cc} \mathbf{F}_{/\tilde{d}} & \mathbf{F}_{/\tilde{d}} \mathbf{A}_1^T + \mathbf{F}_{/\theta} \\ \mathbf{A}_1 \mathbf{F}_{/\tilde{d}} & \mathbf{A}_1 (\mathbf{F}_{/\tilde{d}} \mathbf{A}_1^T + \mathbf{F}_{/\theta}) + \mathbf{M}_{/\tilde{d}} \mathbf{A}_1^T + \mathbf{M}_{/\theta} \\ -\mathbf{F}_{/\tilde{d}} & -\mathbf{F}_{/\tilde{d}} \mathbf{A}_1^T - \mathbf{F}_{/\theta} \\ -\mathbf{A}_2 \mathbf{F}_{/\tilde{d}} & -\mathbf{A}_2 (\mathbf{F}_{/\tilde{d}} \mathbf{A}_1^T + \mathbf{F}_{/\theta}) - \mathbf{M}_{/\tilde{d}} \mathbf{A}_1^T - \mathbf{M}_{/\theta} \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_1 \\ \delta \mathbf{g}_1 \end{array} \right\} \\
& + \left[\begin{array}{cc} -\mathbf{F}_{/\tilde{d}} & -\mathbf{F}_{/\tilde{d}} \mathbf{A}_2^T - \mathbf{F}_{/\theta} \\ -\mathbf{A}_1 \mathbf{F}_{/\tilde{d}} & -\mathbf{A}_1 (\mathbf{F}_{/\tilde{d}} \mathbf{A}_2^T + \mathbf{F}_{/\theta}) - \mathbf{M}_{/\tilde{d}} \mathbf{A}_2^T - \mathbf{M}_{/\theta} \\ \mathbf{F}_{/\tilde{d}} & \mathbf{F}_{/\tilde{d}} \mathbf{A}_2^T + \mathbf{F}_{/\theta} \\ \mathbf{A}_2 \mathbf{F}_{/\tilde{d}} & \mathbf{A}_2 (\mathbf{F}_{/\tilde{d}} \mathbf{A}_2^T + \mathbf{F}_{/\theta}) + \mathbf{M}_{/\tilde{d}} \mathbf{A}_2^T + \mathbf{M}_{/\theta} \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_2 \\ \delta \mathbf{g}_2 \end{array} \right\} \\
& + \left[\begin{array}{cc} \mathbf{0} & \mathbf{F} \times \hat{\mathbf{I}}^T \\ -\hat{\mathbf{I}} \mathbf{F} \times & (\mathbf{A}_1 \mathbf{F} \times + \mathbf{M} \times) \hat{\mathbf{I}}^T - \hat{\mathbf{I}}^T \mathbf{F} \times \mathbf{f}_1 \times - \hat{\mathbf{I}}_{1(d \times F)} \\ \mathbf{0} & -\mathbf{F} \times \hat{\mathbf{I}}^T \\ -\hat{\mathbf{I}}^T \mathbf{F} \times & -(\mathbf{A}_2 \mathbf{F} \times + \mathbf{M} \times) \hat{\mathbf{I}}^T - \hat{\mathbf{I}}^T \mathbf{F} \times \mathbf{f}_1 \times - (\hat{\mathbf{I}}^T)_{1(d \times F)} \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_1 \\ \delta \mathbf{g}_1 \end{array} \right\} \\
& + \left[\begin{array}{cc} \mathbf{0} & \mathbf{F} \times \hat{\mathbf{I}} \\ \hat{\mathbf{I}} \mathbf{F} \times & (\mathbf{A}_1 \mathbf{F} \times + \mathbf{M} \times) \hat{\mathbf{I}} - \hat{\mathbf{I}} \mathbf{F} \times \mathbf{f}_2 \times - \hat{\mathbf{I}}_{2(d \times F)} \\ \mathbf{0} & -\mathbf{F} \times \hat{\mathbf{I}} \\ \hat{\mathbf{I}}^T \mathbf{F} \times & -(\mathbf{A}_2 \mathbf{F} \times + \mathbf{M} \times) \hat{\mathbf{I}} + \hat{\mathbf{I}} \mathbf{F} \times \mathbf{f}_2 \times + (\hat{\mathbf{I}}^T)_{2(d \times F)} \end{array} \right] \left\{ \begin{array}{l} \delta \mathbf{x}_2 \\ \delta \mathbf{g}_2 \end{array} \right\} \\
& = \left\{ \begin{array}{l} \mathbf{F} \\ \mathbf{A}_1 \mathbf{F} + \mathbf{M} \\ -\mathbf{F} \\ -\mathbf{A}_2 \mathbf{F} - \mathbf{M} \end{array} \right\} \tag{9.335}
\end{aligned}$$

with

$$\hat{\mathbf{C}}_1 = \mathbf{C}_1 + \boldsymbol{\omega}_1 \times \tag{9.336a}$$

$$\hat{\mathbf{C}}_2 = \mathbf{C}_2 - \boldsymbol{\omega}_2 \times . \tag{9.336b}$$

Note that

$$\hat{\mathbf{C}}_1 = (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \hat{\mathbf{I}}^T + \boldsymbol{\omega}_1 \times = \boldsymbol{\omega}_1 \times \hat{\mathbf{I}} + \boldsymbol{\omega}_2 \times \hat{\mathbf{I}}^T \quad (9.337a)$$

$$\hat{\mathbf{C}}_2 = (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \hat{\mathbf{I}} - \boldsymbol{\omega}_2 \times = -\boldsymbol{\omega}_1 \times \hat{\mathbf{I}} - \boldsymbol{\omega}_2 \times \hat{\mathbf{I}}^T, \quad (9.337b)$$

so $\hat{\mathbf{C}}_2 = -\hat{\mathbf{C}}_1$.

9.2.5 Deformable Axial Joint

Strain:

$$\boldsymbol{\theta} = \text{ax}(\exp^{-1}(\mathbf{R}_{1h}^T \mathbf{R}_{2h})) \quad (9.338)$$

$$\epsilon = \tilde{\mathbf{e}}_z \cdot \boldsymbol{\theta} \quad (9.339)$$

Strain rate:

$$\dot{\epsilon} = \tilde{\mathbf{e}}_z \cdot \mathbf{R}_{1h}^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \quad (9.340)$$

Virtual perturbation/linearization of strain:

$$\delta\epsilon = \tilde{\mathbf{e}}_z \cdot \mathbf{R}_{1h}^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) = \mathbf{e}_z^T (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta}) \quad (9.341)$$

Linearization of strain rate:

$$\delta\dot{\epsilon} = \tilde{\mathbf{e}}_z \cdot \mathbf{R}_{1h}^T ((\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \boldsymbol{\theta}_{1\delta} + \delta\boldsymbol{\omega}_2 - \delta\boldsymbol{\omega}_1) = \mathbf{e}_z^T ((\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \boldsymbol{\theta}_{1\delta} + \delta\boldsymbol{\omega}_2 - \delta\boldsymbol{\omega}_1) \quad (9.342)$$

Virtual work:

$$\delta\mathcal{L} = \delta\epsilon m(\epsilon, \dot{\epsilon}) = (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta})^T \mathbf{R}_{1h} \tilde{\mathbf{e}}_z m = (\boldsymbol{\theta}_{2\delta} - \boldsymbol{\theta}_{1\delta})^T \mathbf{e}_z m \quad (9.343)$$

Loads (residual contribution):

$$\mathbf{m}_1 = \mathbf{e}_z m \quad (9.344a)$$

$$\mathbf{m}_2 = -\mathbf{e}_z m \quad (9.344b)$$

Linearization:

$$\delta(\mathbf{e}_z m) = -m \mathbf{e}_z \times \boldsymbol{\theta}_{1\delta} + \mathbf{e}_z (m_{/\epsilon} \delta\epsilon + m_{/\dot{\epsilon}} \delta\dot{\epsilon}) \quad (9.345)$$

Jacobian matrix contributions:

$$m \begin{bmatrix} \mathbf{e}_z \times \\ -\mathbf{e}_z \times \end{bmatrix} \boldsymbol{\theta}_{1\delta} + m_{/\epsilon} \begin{bmatrix} \mathbf{e}_z \mathbf{e}_z^T & -\mathbf{e}_z \mathbf{e}_z^T \\ -\mathbf{e}_z \mathbf{e}_z^T & \mathbf{e}_z \mathbf{e}_z^T \end{bmatrix} \begin{Bmatrix} \boldsymbol{\theta}_{1\delta} \\ \boldsymbol{\theta}_{2\delta} \end{Bmatrix} + m_{/\dot{\epsilon}} \left(\begin{bmatrix} \mathbf{e}_z \mathbf{e}_z^T & -\mathbf{e}_z \mathbf{e}_z^T \\ -\mathbf{e}_z \mathbf{e}_z^T & \mathbf{e}_z \mathbf{e}_z^T \end{bmatrix} \begin{Bmatrix} \delta\boldsymbol{\omega}_1 \\ \delta\boldsymbol{\omega}_2 \end{Bmatrix} + \begin{bmatrix} -\mathbf{e}_z \mathbf{e}_z^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \\ \mathbf{e}_z \mathbf{e}_z^T (\boldsymbol{\omega}_2 - \boldsymbol{\omega}_1) \times \end{bmatrix} \boldsymbol{\theta}_{1\delta} \right) = \begin{Bmatrix} \mathbf{e}_z \\ -\mathbf{e}_z \end{Bmatrix} m \quad (9.346)$$

With the updated-updated approximation:

$$m \begin{bmatrix} \mathbf{e}_z \times \\ -\mathbf{e}_z \times \end{bmatrix} \delta\mathbf{g}_1 + m_{/\epsilon} \begin{bmatrix} \mathbf{e}_z \mathbf{e}_z^T & -\mathbf{e}_z \mathbf{e}_z^T \\ -\mathbf{e}_z \mathbf{e}_z^T & \mathbf{e}_z \mathbf{e}_z^T \end{bmatrix} \begin{Bmatrix} \delta\mathbf{g}_1 \\ \delta\mathbf{g}_2 \end{Bmatrix} + m_{/\dot{\epsilon}} \left(\begin{bmatrix} \mathbf{e}_z \mathbf{e}_z^T & -\mathbf{e}_z \mathbf{e}_z^T \\ -\mathbf{e}_z \mathbf{e}_z^T & \mathbf{e}_z \mathbf{e}_z^T \end{bmatrix} \begin{Bmatrix} \delta\dot{\mathbf{g}}_1 \\ \delta\dot{\mathbf{g}}_2 \end{Bmatrix} + \begin{bmatrix} \mathbf{e}_z \mathbf{e}_z^T \boldsymbol{\omega}_2 \times & -\mathbf{e}_z \mathbf{e}_z^T \boldsymbol{\omega}_2 \times \\ -\mathbf{e}_z \mathbf{e}_z^T \boldsymbol{\omega}_2 \times & \mathbf{e}_z \mathbf{e}_z^T \boldsymbol{\omega}_2 \times \end{bmatrix} \begin{Bmatrix} \delta\mathbf{g}_1 \\ \delta\mathbf{g}_2 \end{Bmatrix} \right) \stackrel{\text{uu}}{=} \begin{Bmatrix} \mathbf{e}_z \\ -\mathbf{e}_z \end{Bmatrix} m \quad (9.347)$$

Note: $\mathbf{e}_z \mathbf{e}_z^T \boldsymbol{\omega}_2 \times = \mathbf{e}_z (\mathbf{e}_z \times \boldsymbol{\omega}_2)^T$.

9.3 Viscous Body

This element implements the behavior of a viscous body, namely a force and a moment that depend on the absolute linear and angular velocity of a node, projected in the reference frame of the node. This element allows, for example, to implement the aerodynamics of a flight-mechanics rigid-body model, whose aerodynamic forces and moments depend on the absolute linear and angular velocity, projected in the reference frame of the body, by means of an appropriate constitutive law.

The force and moment are defined as

$$\mathbf{f} = \mathbf{R}\mathbf{R}_h\tilde{\mathbf{f}} \quad (9.348a)$$

$$\mathbf{m} = \mathbf{R}\mathbf{R}_h\tilde{\mathbf{m}} + \mathbf{o} \times \mathbf{f} \quad (9.348b)$$

where

$$\mathbf{o} = \mathbf{R}\tilde{\mathbf{o}}, \quad (9.349)$$

$$\tilde{\mathbf{f}} = \tilde{\mathbf{f}}(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}) \quad (9.350a)$$

$$\tilde{\mathbf{m}} = \tilde{\mathbf{m}}(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}) \quad (9.350b)$$

and

$$\tilde{\mathbf{v}} = \mathbf{R}_h^T \mathbf{R}^T (\dot{\mathbf{x}} + \boldsymbol{\omega} \times \mathbf{o}) \quad (9.351a)$$

$$\tilde{\boldsymbol{\omega}} = \mathbf{R}_h^T \mathbf{R}^T \boldsymbol{\omega}. \quad (9.351b)$$

The linearization of the force and moment yields

$$\delta\tilde{\mathbf{f}} = \tilde{\mathbf{f}}_{/\tilde{\mathbf{v}}} \delta\tilde{\mathbf{v}} + \tilde{\mathbf{f}}_{/\tilde{\boldsymbol{\omega}}} \delta\tilde{\boldsymbol{\omega}} \quad (9.352a)$$

$$\delta\tilde{\mathbf{m}} = \tilde{\mathbf{m}}_{/\tilde{\mathbf{v}}} \delta\tilde{\mathbf{v}} + \tilde{\mathbf{m}}_{/\tilde{\boldsymbol{\omega}}} \delta\tilde{\boldsymbol{\omega}} \quad (9.352b)$$

with

$$\delta\tilde{\mathbf{v}} = \mathbf{R}_h^T \mathbf{R}^T (\delta\dot{\mathbf{x}} + \delta\boldsymbol{\omega} \times \mathbf{o} + \boldsymbol{\omega} \times \boldsymbol{\theta}_\delta \times \mathbf{o}) \stackrel{\text{uu}}{=} \mathbf{R}_h^T \mathbf{R}^T (\delta\dot{\mathbf{x}} - \mathbf{o} \times \delta\dot{\mathbf{g}} + (\mathbf{o} \times \boldsymbol{\omega}) \times \delta\mathbf{g}) \quad (9.353a)$$

$$\delta\tilde{\boldsymbol{\omega}} = \mathbf{R}_h^T \mathbf{R}^T (\delta\boldsymbol{\omega} + \boldsymbol{\omega} \times \boldsymbol{\theta}_\delta) \stackrel{\text{uu}}{=} \mathbf{R}_h^T \mathbf{R}^T \delta\dot{\mathbf{g}} \quad (9.353b)$$

The linearization of the nodal forces and moments yields

$$\delta\mathbf{f} = \boldsymbol{\theta}_\delta \times \mathbf{f} + \mathbf{R}\mathbf{R}_h\delta\tilde{\mathbf{f}} \quad (9.354a)$$

$$\delta\mathbf{m} = \boldsymbol{\theta}_\delta \times \mathbf{m} + \mathbf{R}\mathbf{R}_h\delta\tilde{\mathbf{m}} + \mathbf{o} \times \mathbf{R}\mathbf{R}_h\delta\tilde{\mathbf{f}}, \quad (9.354b)$$

namely

$$\begin{aligned} & \left[\begin{array}{cc} \mathbf{f}_{/\tilde{\mathbf{v}}} & \mathbf{f}_{/\tilde{\boldsymbol{\omega}}} - \mathbf{f}_{/\tilde{\mathbf{v}}} \mathbf{o} \times \\ \mathbf{m}_{/\tilde{\mathbf{v}}} + \mathbf{o} \times \mathbf{f}_{/\tilde{\mathbf{v}}} & \mathbf{m}_{/\tilde{\boldsymbol{\omega}}} - \mathbf{m}_{/\tilde{\mathbf{v}}} \mathbf{o} \times + \mathbf{o} \times (\mathbf{f}_{/\tilde{\boldsymbol{\omega}}} - \mathbf{f}_{/\tilde{\mathbf{v}}} \mathbf{o} \times) \end{array} \right] \left\{ \begin{array}{c} \delta\dot{\mathbf{x}} \\ \delta\boldsymbol{\omega} \end{array} \right\} \\ & + \left[\begin{array}{c} \mathbf{f}_{/\tilde{\boldsymbol{\omega}}} \boldsymbol{\omega} \times - \mathbf{f}_{/\tilde{\mathbf{v}}} \boldsymbol{\omega} \times \mathbf{o} \times \\ \mathbf{m}_{/\tilde{\boldsymbol{\omega}}} \boldsymbol{\omega} \times - \mathbf{m}_{/\tilde{\mathbf{v}}} \boldsymbol{\omega} \times \mathbf{o} \times + \mathbf{o} \times (\mathbf{f}_{/\tilde{\boldsymbol{\omega}}} \boldsymbol{\omega} \times - \mathbf{f}_{/\tilde{\mathbf{v}}} \boldsymbol{\omega} \times \mathbf{o} \times) \end{array} \right] \boldsymbol{\theta}_\delta \\ & \quad + \left[\begin{array}{c} -\mathbf{f} \times \\ -\mathbf{m} \times \end{array} \right] \boldsymbol{\theta}_\delta = \left\{ \begin{array}{c} \delta\mathbf{f} \\ \delta\mathbf{m} \end{array} \right\} \end{aligned} \quad (9.355)$$

where

$$\mathbf{f}_{/\tilde{\mathbf{v}}} = \mathbf{R}\mathbf{R}_h\tilde{\mathbf{f}}_{/\tilde{\mathbf{v}}}\mathbf{R}_h^T\mathbf{R}^T \quad (9.356a)$$

$$\mathbf{f}_{/\tilde{\omega}} = \mathbf{R}\mathbf{R}_h\tilde{\mathbf{f}}_{/\tilde{\omega}}\mathbf{R}_h^T\mathbf{R}^T \quad (9.356b)$$

$$\mathbf{m}_{/\tilde{\mathbf{v}}} = \mathbf{R}\mathbf{R}_h\tilde{\mathbf{m}}_{/\tilde{\mathbf{v}}}\mathbf{R}_h^T\mathbf{R}^T \quad (9.356c)$$

$$\mathbf{m}_{/\tilde{\omega}} = \mathbf{R}\mathbf{R}_h\tilde{\mathbf{m}}_{/\tilde{\omega}}\mathbf{R}_h^T\mathbf{R}^T \quad (9.356d)$$

The updated-updated approximation yields

$$\begin{bmatrix} \mathbf{f}_{/\tilde{\mathbf{v}}} \\ \mathbf{m}_{/\tilde{\mathbf{v}}} + \mathbf{o} \times \mathbf{f}_{/\tilde{\mathbf{v}}} \\ + \left[\begin{array}{c} \mathbf{f}_{/\tilde{\mathbf{v}}} - \mathbf{f}_{/\tilde{\mathbf{v}}} \mathbf{o} \times \\ (\mathbf{m}_{/\tilde{\mathbf{v}}} + \mathbf{o} \times \mathbf{f}_{/\tilde{\mathbf{v}}}) (\mathbf{o} \times \mathbf{\omega}) \times \end{array} \right] \end{bmatrix} \begin{Bmatrix} \delta \dot{\mathbf{x}} \\ \delta \dot{\mathbf{g}} \end{Bmatrix} + \left[\begin{array}{c} \mathbf{f}_{/\tilde{\omega}} - \mathbf{f}_{/\tilde{\mathbf{v}}} \mathbf{o} \times \\ (\mathbf{m}_{/\tilde{\omega}} - \mathbf{m}_{/\tilde{\mathbf{v}}} \mathbf{o} \times + \mathbf{o} \times (\mathbf{f}_{/\tilde{\omega}} - \mathbf{f}_{/\tilde{\mathbf{v}}} \mathbf{o} \times) \end{array} \right] \begin{Bmatrix} \delta \mathbf{f} \\ \delta \mathbf{m} \end{Bmatrix} = \left[\begin{array}{c} \mathbf{f} \times \\ \mathbf{m} \times \end{array} \right] \quad (9.357)$$

9.4 Modal Element

9.4.1 Kinematics

Position of an arbitrary point P

$$\mathbf{x}_P = \mathbf{x}_0 + \mathbf{f}_P + \mathbf{u}_P \quad (9.358)$$

where \mathbf{x}_0 is the position of the point that describes the global motion of the body, \mathbf{f}_P is the relative position of the point when the body is undeformed, and \mathbf{u}_P is the relative displacement of the point when the body is deformed.

It can be rewritten as

$$\mathbf{x}_P = \mathbf{x}_0 + \mathbf{R}_0 \left(\tilde{\mathbf{f}}_P + \tilde{\mathbf{u}}_P \right) \quad (9.359)$$

where \mathbf{R}_0 is the global orientation matrix of the body, and the *tilde* ($\tilde{\cdot}$) indicates entities expressed in the reference frame attached to the body.

The deformation of the body is expressed by a linear combination of M displacement (and rotation, for those models that consider them, like beam trusses) shapes

$$\tilde{\mathbf{u}}_P = \sum_{j=1,M} \mathbf{U}_{Pj} q_j = \mathbf{U}_P \mathbf{q} \quad (9.360)$$

where \mathbf{U}_{Pj} is the vector containing the components of the j -th displacement shape related to point P , and q_j is the j -th mode multiplier.

The orientation of the generic point P is

$$\mathbf{R}_P = \mathbf{R}_0 \tilde{\mathbf{R}}_P \quad (9.361)$$

and, assuming a representation of the relative orientation by a linear combination of rotation shapes

$$\tilde{\phi} = \sum_{j=1,M} \mathbf{V}_{Pj} q_j = \mathbf{V}_P \mathbf{q} \quad (9.362)$$

it results in a linearized orientation

$$\mathbf{R}_P \cong \mathbf{R}_0 (\mathbf{I} + (\mathbf{V}_P \mathbf{q}) \times) \quad (9.363)$$

which is no longer orthogonal, because of matrix

$$\tilde{\mathbf{R}}_P = \mathbf{I} + (\mathbf{V}_P \mathbf{q}) \times \quad (9.364)$$

which represents a linearized rotation.

The first and second derivatives of position and orientation yield:

$$\dot{\mathbf{x}}_P = \dot{\mathbf{x}}_0 + \boldsymbol{\omega}_0 \times \mathbf{R}_0 (\tilde{\mathbf{f}}_P + \mathbf{U}_P \mathbf{q}) + \mathbf{R}_0 \mathbf{U}_P \dot{\mathbf{q}} \quad (9.365a)$$

$$\boldsymbol{\omega}_P = \boldsymbol{\omega}_0 + \mathbf{R}_0 \mathbf{V}_P \dot{\mathbf{q}} \quad (9.365b)$$

$$\begin{aligned} \ddot{\mathbf{x}}_P &= \ddot{\mathbf{x}}_0 + \dot{\boldsymbol{\omega}}_0 \times \mathbf{R}_0 (\tilde{\mathbf{f}}_P + \mathbf{U}_P \mathbf{q}) + \boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times \mathbf{R}_0 (\tilde{\mathbf{f}}_P + \mathbf{U}_P \mathbf{q}) \\ &\quad + 2\boldsymbol{\omega}_0 \times \mathbf{R}_0 \mathbf{U}_P \dot{\mathbf{q}} + \mathbf{R}_0 \mathbf{U}_P \ddot{\mathbf{q}} \end{aligned} \quad (9.365c)$$

$$\dot{\boldsymbol{\omega}}_P = \dot{\boldsymbol{\omega}}_0 + \boldsymbol{\omega}_0 \times \mathbf{R}_0 \mathbf{V}_P \dot{\mathbf{q}} + \mathbf{R}_0 \mathbf{V}_P \ddot{\mathbf{q}} \quad (9.365d)$$

The virtual perturbation of the position and orientation of the generic point P are:

$$\delta \mathbf{x}_P = \delta \mathbf{x}_0 + \delta \boldsymbol{\phi}_0 \times \mathbf{R}_0 (\tilde{\mathbf{f}}_P + \mathbf{U}_P \mathbf{q}) + \mathbf{R}_0 \mathbf{U}_P \delta \mathbf{q} \quad (9.366a)$$

$$\delta \boldsymbol{\phi}_P = \delta \boldsymbol{\phi}_0 + \mathbf{R}_0 \mathbf{V}_P \delta \mathbf{q} \quad (9.366b)$$

Without significant losses in generality, from now on it is assumed that the structure of the problem is given in form of lumped inertia parameters in specific points, corresponding to FEM nodes, and that the position of each node corresponds to the center of mass of each lumped mass. A model made of N FEM nodes is considered. The nodal mass of the i -th FEM node is

$$\mathbf{M}_i = \begin{bmatrix} m_i \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_i \end{bmatrix} \quad (9.367)$$

There is no strict requirement for matrix \mathbf{J}_i to be diagonal.

The inertia forces and moments acting on each FEM node are:

$$\mathbf{F}_i = -m_i \ddot{\mathbf{x}}_i \quad (9.368a)$$

$$\mathbf{C}_i = -\mathbf{R}_i \mathbf{J}_i \mathbf{R}_i^T \dot{\boldsymbol{\omega}}_i \quad (9.368b)$$

and the virtual work done by the inertia forces is

$$\delta L = \sum_{i=1,N} (\delta \mathbf{x}_i^T \mathbf{F}_i + \delta \boldsymbol{\phi}_i^T \mathbf{C}_i) \quad (9.369)$$

which results in

$$\delta L = \left\{ \begin{array}{c} \delta \mathbf{x}_0 \\ \delta \boldsymbol{\phi}_0 \\ \delta \mathbf{q} \end{array} \right\}^T \left(\left[\begin{array}{ccc} \mathbf{M}_{xx} & \mathbf{M}_{x\phi} & \mathbf{M}_{xq} \\ \mathbf{M}_{\phi x} & \mathbf{M}_{\phi\phi} & \mathbf{M}_{\phi q} \\ \text{sym.} & \mathbf{M}_{q\phi} & \mathbf{M}_{qq} \end{array} \right] \left\{ \begin{array}{c} \ddot{\mathbf{x}}_0 \\ \dot{\boldsymbol{\omega}}_0 \\ \ddot{\mathbf{q}} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{F}_x \\ \mathbf{F}_\phi \\ \mathbf{F}_q \end{array} \right\} \right) \quad (9.370)$$

with

$$\mathbf{M}_{xx} = \mathbf{I} \sum_{i=1,N} m_i \quad (9.371\text{a})$$

$$\mathbf{M}_{x\phi} = \mathbf{R}_0 \sum_{i=1,N} m_i (\tilde{\mathbf{f}}_i + \mathbf{U}_i \mathbf{q}) \times^T \mathbf{R}_0^T \quad (9.371\text{b})$$

$$\mathbf{M}_{xq} = \mathbf{R}_0 \sum_{i=1,N} m_i \mathbf{U}_i \quad (9.371\text{c})$$

$$\mathbf{M}_{\phi\phi} = \mathbf{R}_0 \sum_{i=1,N} \left(m_i (\tilde{\mathbf{f}}_i + \mathbf{U}_i \mathbf{q}) \times (\tilde{\mathbf{f}}_i + \mathbf{U}_i \mathbf{q}) \times^T + \tilde{\mathbf{R}}_i \mathbf{J}_i \tilde{\mathbf{R}}_i^T \right) \mathbf{R}_0^T \quad (9.371\text{d})$$

$$\mathbf{M}_{\phi q} = \mathbf{R}_0 \sum_{i=1,N} \left(m_i (\tilde{\mathbf{f}}_i + \mathbf{U}_i \mathbf{q}) \times \mathbf{U}_i + \tilde{\mathbf{R}}_i \mathbf{J}_i \tilde{\mathbf{R}}_i^T \mathbf{V}_i \right) \quad (9.371\text{e})$$

$$\mathbf{M}_{qq} = \sum_{i=1,N} \left(m_i \mathbf{U}_i^T \mathbf{U}_i + \mathbf{V}_i^T \tilde{\mathbf{R}}_i \mathbf{J}_i \tilde{\mathbf{R}}_i^T \mathbf{V}_i \right) \quad (9.371\text{f})$$

$$\mathbf{F}_x = \sum_{i=1,N} m_i \left(\boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times \mathbf{R}_0 (\tilde{\mathbf{f}}_i + \mathbf{U}_i \mathbf{q}) + 2\boldsymbol{\omega}_0 \times \mathbf{R}_0 \mathbf{U}_i \dot{\mathbf{q}} \right) \quad (9.371\text{g})$$

$$\begin{aligned} \mathbf{F}_\phi = & \sum_{i=1,N} \mathbf{R}_0 \left(m_i (\tilde{\mathbf{f}}_i + \mathbf{U}_i \mathbf{q}) \times (\boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times \mathbf{R}_0 (\tilde{\mathbf{f}}_i + \mathbf{U}_i \mathbf{q}) + 2\boldsymbol{\omega}_0 \times \mathbf{R}_0 \mathbf{U}_i \dot{\mathbf{q}}) \right. \\ & \left. + \tilde{\mathbf{R}}_i \mathbf{J}_i \tilde{\mathbf{R}}_i^T \mathbf{R}_0^T \boldsymbol{\omega}_0 \times \mathbf{R}_0 \mathbf{V}_i \dot{\mathbf{q}} \right) \end{aligned} \quad (9.371\text{h})$$

$$\begin{aligned} \mathbf{F}_q = & \sum_{i=1,N} \left(m_i \mathbf{U}_i^T \mathbf{R}_0^T \left(\boldsymbol{\omega}_0 \times \boldsymbol{\omega}_0 \times \mathbf{R}_0 (\tilde{\mathbf{f}}_i + \mathbf{U}_i \mathbf{q}) + 2\boldsymbol{\omega}_0 \times \mathbf{R}_0 \mathbf{U}_i \dot{\mathbf{q}} \right. \right. \\ & \left. \left. + \mathbf{V}_i^T \tilde{\mathbf{R}}_i \mathbf{J}_i \tilde{\mathbf{R}}_i^T \mathbf{R}_0^T \boldsymbol{\omega}_0 \times \mathbf{R}_0 \mathbf{V}_i \dot{\mathbf{q}} \right) \right) \end{aligned} \quad (9.371\text{i})$$

The \mathbf{M}_{jk} terms can be rewritten to highlight contributions of order 0, 1, and higher:

$$\mathbf{M}_{xx} = \mathbf{I} \left(\sum_{i=1,N} m_i \right) \quad (9.372a)$$

$$\mathbf{M}_{x\phi} = \mathbf{R}_0 \left(\left(\sum_{i=1,N} m_i \tilde{\mathbf{f}}_i \right) + \left(\left(\sum_{i=1,N} m_i \mathbf{U}_i \right) \mathbf{q} \right) \right) \times^T \mathbf{R}_0^T \quad (9.372b)$$

$$\mathbf{M}_{xq} = \mathbf{R}_0 \left(\sum_{i=1,N} m_i \mathbf{U}_i \right) \quad (9.372c)$$

$$\begin{aligned} \mathbf{M}_{\phi\phi} = & \mathbf{R}_0 \left(\sum_{i=1,N} \left(m_i \tilde{\mathbf{f}}_i \times \tilde{\mathbf{f}}_i \times^T + \mathbf{J}_i \right) \right. \\ & + \sum_{i=1,N} \left(m_i \tilde{\mathbf{f}}_i \times (\mathbf{U}_i \mathbf{q}) \times^T + m_i (\mathbf{U}_i \mathbf{q}) \times \tilde{\mathbf{f}}_i \times^T + \mathbf{J}_i (\mathbf{V}_i \mathbf{q}) \times^T + (\mathbf{V}_i \mathbf{q}) \times \mathbf{J}_i \right) \\ & \left. + \sum_{i=1,N} \left(m_i (\mathbf{U}_i \mathbf{q}) \times (\mathbf{U}_i \mathbf{q}) \times^T + (\mathbf{V}_i \mathbf{q}) \times \mathbf{J}_i (\mathbf{V}_i \mathbf{q}) \times^T \right) \right) \mathbf{R}_0^T \end{aligned} \quad (9.372d)$$

$$\begin{aligned} \mathbf{M}_{\phi q} = & \mathbf{R}_0 \left(\sum_{i=1,N} \left(m_i \tilde{\mathbf{f}}_i \times \mathbf{U}_i + \mathbf{J}_i \mathbf{V}_i \right) \right. \\ & + \sum_{i=1,N} \left(m_i (\mathbf{U}_i \mathbf{q}) \times \mathbf{U}_i + \mathbf{J}_i (\mathbf{V}_i \mathbf{q}) \times^T \mathbf{V}_i + (\mathbf{V}_i \mathbf{q}) \times \mathbf{J}_i \mathbf{V}_i \right) \\ & \left. + \sum_{i=1,N} (\mathbf{V}_i \mathbf{q}) \times \mathbf{J}_i (\mathbf{V}_i \mathbf{q}) \times^T \mathbf{V}_i \right) \end{aligned} \quad (9.372e)$$

$$\begin{aligned} \mathbf{M}_{qq} = & \sum_{i=1,N} \left(m_i \mathbf{U}_i^T \mathbf{U}_i + \mathbf{V}_i^T \mathbf{J}_i \mathbf{V}_i \right) \\ & + \sum_{i=1,N} \left(\mathbf{V}_i^T \mathbf{J}_i (\mathbf{V}_i \mathbf{q}) \times^T \mathbf{V}_i + \mathbf{V}_i^T (\mathbf{V}_i \mathbf{q}) \times \mathbf{J}_i \mathbf{V}_i \right) \\ & + \sum_{i=1,N} \mathbf{V}_i^T (\mathbf{V}_i \mathbf{q}) \times \mathbf{J}_i (\mathbf{V}_i \mathbf{q}) \times^T \mathbf{V}_i \end{aligned} \quad (9.372f)$$

9.4.2 Physics: Orthogonality

Some noteworthy entities appear in the above equations, which may partially simplify under special circumstances.

The overall mass of the body

$$m = \sum_{i=1,N} m_i \quad (9.373)$$

The static (first order) inertia moment

$$\mathbf{S}_{x\phi} = \sum_{i=1,N} m_i \tilde{\mathbf{f}}_i \quad (9.374)$$

vanishes if point \mathbf{x}_0 is the center of mass of the undeformed body.

Similarly, the static (first order) inertia moment computed with the modal displacement shapes

$$\mathbf{S}_{xq} = \sum_{i=1,N} m_i \mathbf{U}_i \quad (9.375)$$

vanishes if the mode shapes have been inertially decoupled from the rigid body displacements. In fact, the decoupling of the rigid and the deformable modes is expressed by

$$\begin{aligned} \sum_{i=1,N} \mathbf{x}_r^T m_i \mathbf{U}_i &= \\ \mathbf{x}_r^T \sum_{i=1,N} m_i \mathbf{U}_i &= 0 \end{aligned} \quad (9.376)$$

where \mathbf{x}_r^T describes three independent rigid translations, which, for the arbitrariness of \mathbf{x}_r , implies the above Equation (9.375).

In the same case, also the zero-order terms of the coupling between the rigid body rotations and the modal variables,

$$\mathbf{S}_{\phi q} = \sum_{i=1,N} \left(m_i \tilde{\mathbf{f}}_i \times \mathbf{U}_i + \mathbf{J}_i \mathbf{V}_i \right) \quad (9.377)$$

also vanishes. In fact, the decoupling of the rigid and the deformable modes is expressed by

$$\begin{aligned} \sum_{i=1,N} \left(m_i \phi_r^T \tilde{\mathbf{f}}_i \times \mathbf{U}_i + \phi_r^T \mathbf{J}_i \mathbf{V}_i \right) &= \\ \phi_r^T \sum_{i=1,N} \left(m_i \tilde{\mathbf{f}}_i \times \mathbf{U}_i + \mathbf{J}_i \mathbf{V}_i \right) &= 0 \end{aligned} \quad (9.378)$$

where ϕ_r^T describes three independent rigid rotations, and $\phi_r^T \tilde{\mathbf{f}}_i \times$ describes the corresponding displacements, which, for the arbitrariness of ϕ_r , implies the above Equation (9.377).

The second order inertia moment is

$$\mathbf{J} = \sum_{i=1,N} \left(m_i \tilde{\mathbf{f}}_i \times \tilde{\mathbf{f}}_i \times^T + \mathbf{J}_i \right) \quad (9.379)$$

It results in a diagonal matrix if the orientation of the body is aligned with the principal inertia axes.

The modal mass matrix is

$$\mathbf{m} = \sum_{i=1,N} \left(m_i \mathbf{U}_i^T \mathbf{U}_i + \mathbf{V}_i^T \mathbf{J}_i \mathbf{V}_i \right) \quad (9.380)$$

It is diagonal if only the normal modes are considered.

9.4.3 Simplifications

The problem, as stated up to now, already contains some simplifications. First of all, those related to the lumped inertia model of a continuum; moreover, those related to the mode superposition to describe the straining of the body which, in the case of the FEM node rotation, yields a non-orthogonal linearized rotation matrix.

Further simplifications are usually accepted in common modeling practice, where some of the higher order terms are simply discarded.

When only the 0-th order coefficients are used in matrices M_{uv} , the dynamics of the body are written referred to the undeformed shape. This approximation can be quite drastic, but in some cases it may be reasonable, if the reference straining, represented by \mathbf{q} , remains very small throughout the simulation. This approximation is also required when the only available data are the global inertia properties (e.g. m , the position of the center of mass and the inertia matrix \mathbf{J}), and the modal mass matrix \mathbf{m} .

More refined approximations include higher order terms: for example, the first and second order contributions illustrated before. This corresponds to using finer and finer descriptions of the inertia properties of the system, corresponding to computing the inertia properties in the deformed condition with first and second order accuracy, respectively.

9.4.4 Invariants

The dynamics of the deformed body can be written without any detailed knowledge of the mass distribution, provided some aggregate information can be gathered in so-called *invariants*. They are:

1. Total mass (scalar)

$$\mathcal{I}_1 = \sum_{i=1,N} m_i \quad (9.381)$$

where m_i is the mass of the i -th FEM node⁴.

2. Static moment (matrix 3×1)

$$\mathcal{I}_2 = \sum_{i=1,N} m_i \tilde{\mathbf{f}}_i \quad (9.382)$$

3. Static coupling between rigid body and FEM node displacements (matrix $3 \times M$)

$$\mathcal{I}_3 = \sum_{i=1,N} m_i \mathbf{U}_i \quad (9.383)$$

where the portion related to the k -th mode is computed by summation of the contribute of each FEM node, obtained by multiplying the FEM node mass m_i by the three components of the modal displacement \mathbf{U}_{ik} of the k -th mode.

4. Static coupling between rigid body rotations and FEM node displacements (matrix $3 \times M$)

$$\mathcal{I}_4 = \sum_{i=1,N} \left(m_i \tilde{\mathbf{f}}_i \times \mathbf{U}_i + \mathbf{J}_i \mathbf{V}_i \right) \quad (9.384)$$

where the portion related to the k -th mode is computed by summation of the contribute of each FEM node, obtained by multiplying the FEM node mass m_i by the cross product of the FEM node position $\tilde{\mathbf{f}}_i$ and the three components of the modal displacement \mathbf{U}_{ik} of the k -th mode.

5. Static coupling between FEM node displacements (matrix $3 \times M \times M$)

$$\mathcal{I}_{5j} = \sum_{i=1,N} m_i \mathbf{U}_{ij} \times \mathbf{U}_i \quad (9.385)$$

where the portion related to the j -th mode is computed by summation of the contribute of each FEM node, obtained by multiplying the FEM node mass m_i by the cross product of the three components of the FEM node j -th modal displacement \mathbf{U}_{ij} and the three components of the k -th modal displacement \mathbf{U}_{ik} .

⁴Although the input format, because of NASTRAN legacy, allows each global direction to have a separate mass value, invariants assume that the same value is given, and only use the one associated to component 1.

6. Modal mass matrix (matrix $M \times M$)

$$\mathcal{I}_6 = \sum_{i=1,N} (m_i \mathbf{U}_i^T \mathbf{U}_i + \mathbf{V}_i^T \mathbf{J}_i \mathbf{V}_i) \quad (9.386)$$

7. Inertia matrix (matrix 3×3)

$$\mathcal{I}_7 = \sum_{i=1,N} (m_i \tilde{\mathbf{f}}_i \times \tilde{\mathbf{f}}_i \times^T + \mathbf{J}_i) \quad (9.387)$$

8. (matrix $3 \times M \times 3$)

$$\mathcal{I}_{8j} = \sum_{i=1,N} m_i \tilde{\mathbf{f}}_i \times \mathbf{U}_{ij} \times^T \quad (9.388)$$

9. (matrix $3 \times M \times M \times 3$)

$$\mathcal{I}_{9jk} = \sum_{i=1,N} m_i \mathbf{U}_{ij} \times \mathbf{U}_{ik} \times \quad (9.389)$$

10. (matrix $3 \times M \times 3$)

$$\mathcal{I}_{10j} = \sum_{i=1,N} \mathbf{V}_{ij} \times \mathbf{J}_i \quad (9.390)$$

11. (matrix $3 \times M$)

$$\mathcal{I}_{11} = \sum_{i=1,N} \mathbf{J}_i \mathbf{V}_i \quad (9.391)$$

Using the invariants, the contributions to the inertia matrix of the body become

$$\mathbf{M}_{xx} = \mathbf{I} \mathcal{I}_1 \quad (9.392a)$$

$$\mathbf{M}_{x\phi} = \mathbf{R}_0 (\mathcal{I}_2 + \mathcal{I}_3 \mathbf{q}) \times \mathbf{R}_0^T \quad (9.392b)$$

$$\mathbf{M}_{xq} = \mathbf{R}_0 \mathcal{I}_3 \quad (9.392c)$$

$$\mathbf{M}_{\phi\phi} = \mathbf{R}_0 (\mathcal{I}_7 + (\mathcal{I}_{8j} + \mathcal{I}_{8j}^T) q_j + \mathcal{I}_{9jk} q_j q_k) \mathbf{R}_0^T \quad (9.392d)$$

$$\begin{aligned} \mathbf{M}_{\phi q} = \mathbf{R}_0 & \left(\mathcal{I}_4 + \mathcal{I}_{5j} q_j + \sum_{i=1,N} (\mathbf{J}_i (\mathbf{V}_i \mathbf{q}) \times^T \mathbf{V}_i + (\mathbf{V}_i \mathbf{q}) \times \mathbf{J}_i \mathbf{V}_i) \right. \\ & \left. + \sum_{i=1,N} (\mathbf{V}_i \mathbf{q}) \times \mathbf{J}_i (\mathbf{V}_i \mathbf{q}) \times^T \mathbf{V}_i \right) \end{aligned} \quad (9.392e)$$

$$\begin{aligned} \mathbf{M}_{qq} = \mathcal{I}_6 & + \sum_{i=1,N} (\mathbf{V}_i^T \mathbf{J}_i (\mathbf{V}_i \mathbf{q}) \times^T \mathbf{V}_i + \mathbf{V}_i^T (\mathbf{V}_i \mathbf{q}) \times \mathbf{J}_i \mathbf{V}_i) \\ & + \sum_{i=1,N} \mathbf{V}_i^T (\mathbf{V}_i \mathbf{q}) \times \mathbf{J}_i (\mathbf{V}_i \mathbf{q}) \times^T \mathbf{V}_i \end{aligned} \quad (9.392f)$$

where summation over repeated indices is assumed. The remaining summation terms could be also cast into some invariant form; however, in common practice (e.g. in ADAMS) they are simply neglected, under the assumption that the finer the discretization, the smaller the FEM node inertia, so that linear and quadratic terms in the nodal rotation become reasonably small, yielding

$$\mathbf{M}_{xx} = \mathbf{I} \mathcal{I}_1 \quad (9.393a)$$

$$\mathbf{M}_{x\phi} = \mathbf{R}_0 (\mathcal{I}_2 + \mathcal{I}_3 \mathbf{q}) \times \mathbf{R}_0^T \quad (9.393b)$$

$$\mathbf{M}_{xq} = \mathbf{R}_0 \mathcal{I}_3 \quad (9.393c)$$

$$\mathbf{M}_{\phi\phi} = \mathbf{R}_0 (\mathcal{I}_7 + (\mathcal{I}_{8j} + \mathcal{I}_{8j}^T) q_j + \mathcal{I}_{9jk} q_j q_k) \mathbf{R}_0^T \quad (9.393d)$$

$$\mathbf{M}_{\phi q} = \mathbf{R}_0 (\mathcal{I}_4 + \mathcal{I}_{5j} q_j) \quad (9.393e)$$

$$\mathbf{M}_{qq} = \mathcal{I}_6 \quad (9.393f)$$

In some cases, the only remaining quadratic term in \mathcal{I}_{9jk} is neglected as well.

9.4.5 Interfacing

The basic interface between the FEM and the multibody world occurs by clamping regular multibody nodes to selected nodes on the FEM mesh. Whenever more sophisticated interfacing is required, for example connecting a multibody node to a combination of FEM nodes, an FEM node equivalent to the desired aggregate of nodes should either be prepared at the FEM side, for example by means of RBEs, or at the FEM database side, for example by averaging existing mode shapes according to the desired pattern, into an equivalent FEM node⁵.

The clamping is imposed by means of a coincidence and a parallelism constraint between the locations and the orientations of the two points: the multibody node N and the FEM node P , according to the expressions

$$\mathbf{x}_N + \mathbf{f}_N = \mathbf{x}_P \quad (9.394)$$

$$\text{ax}(\exp^{-1}(\mathbf{R}_N^T \mathbf{R}_P)) = \mathbf{0} \quad (9.395)$$

which becomes

$$\mathbf{x}_N + \mathbf{R}_N \tilde{\mathbf{f}}_N - \mathbf{x}_0 - \mathbf{R}_0 (\tilde{\mathbf{f}}_P + \mathbf{U}_P \mathbf{q}) = \mathbf{0} \quad (9.396)$$

$$\text{ax}(\exp^{-1}(\mathbf{R}_N^T \mathbf{R}_0 (\mathbf{I} + (\mathbf{V}_P \mathbf{q}) \times))) = \mathbf{0} \quad (9.397)$$

The reaction forces exchanged are $\boldsymbol{\lambda}$ in the global frame, while the reaction moments are $\mathbf{R}_N \boldsymbol{\alpha}$ in the reference frame of node N :

$$\mathbf{F}_N = \boldsymbol{\lambda} \quad (9.398)$$

$$\mathbf{M}_N = \mathbf{f}_N \times \boldsymbol{\lambda} + \mathbf{R}_N \boldsymbol{\alpha} \quad (9.399)$$

$$\mathbf{F}_P = -\boldsymbol{\lambda} \quad (9.400)$$

$$\mathbf{M}_P = -\mathbf{R}_N \boldsymbol{\alpha} \quad (9.401)$$

The force and the moment apply on the rigid body displacement and rotation, and on the modal equations as well, according to

$$\begin{aligned} \delta \mathbf{x}_P^T \mathbf{F}_P \\ + \delta \phi_P^T \mathbf{M}_P \end{aligned} = \left\{ \begin{array}{c} \delta \mathbf{x}_0 \\ \delta \phi_0 \\ \delta \mathbf{q} \end{array} \right\}^T \left\{ \begin{array}{c} -\boldsymbol{\lambda} \\ -(\mathbf{R}_0 (\tilde{\mathbf{f}}_P + \mathbf{U}_P \mathbf{q})) \times \boldsymbol{\lambda} - \mathbf{R}_N \boldsymbol{\alpha} \\ -\mathbf{U}_P^T \mathbf{R}_0^T \boldsymbol{\lambda} - \mathbf{V}_P^T \mathbf{R}_0^T \mathbf{R}_N \boldsymbol{\alpha} \end{array} \right\} \quad (9.402)$$

⁵For example, to constrain the displacement of a FEM node P that represents the weighing of the displacement of a set of FEM nodes according to a constant weighing matrix $\mathbf{W}_P \in \mathbb{R}^{3n \times 3}$, simply use $\mathbf{U}_P = \mathbf{W}_P^T \mathbf{U}$.

The linearization of the constraint yields

$$\begin{bmatrix} -\mathbf{I} & \mathbf{f}_N \times & \mathbf{I} & -\left(\mathbf{R}_0 \left(\tilde{\mathbf{f}}_P + \mathbf{U}_P \mathbf{q} \right) \right) \times & \mathbf{R}_0 \mathbf{U}_P \\ \mathbf{0} & -\boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_N^T & \mathbf{0} & \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_N^T & \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \mathbf{R}_N^T \mathbf{R}_0 \mathbf{V}_P \end{bmatrix} \begin{Bmatrix} \delta \mathbf{x}_N \\ \delta \mathbf{g}_N \\ \delta \mathbf{x}_0 \\ \delta \mathbf{g}_0 \\ \delta \mathbf{q} \end{Bmatrix} = \begin{Bmatrix} (9.396) \\ (9.397) \end{Bmatrix} \quad (9.403)$$

Note that $\boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1} \cong \mathbf{I}$ since $\boldsymbol{\theta} \rightarrow \mathbf{0}$ when the constraint is satisfied. The linearization of forces and moments yields

$$\begin{aligned} & \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{\lambda} \times \mathbf{f}_N \times + (\mathbf{R}_N \boldsymbol{\alpha}) \times \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -(\mathbf{R}_N \boldsymbol{\alpha}) \times \\ \mathbf{0} & -\mathbf{V}_P^T \mathbf{R}_0^T (\mathbf{R}_N \boldsymbol{\alpha}) \times \end{bmatrix} \begin{Bmatrix} \delta \mathbf{x}_N \\ \delta \mathbf{g}_N \end{Bmatrix} \\ & + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda} \times \left(\mathbf{R}_0 \left(\tilde{\mathbf{f}}_P + \mathbf{U}_P \mathbf{q} \right) \right) \times & -\boldsymbol{\lambda} \times \mathbf{R}_0 \mathbf{U}_P \\ \mathbf{0} & \mathbf{U}_P^T \mathbf{R}_0^T \boldsymbol{\lambda} \times + \mathbf{V}_P^T \mathbf{R}_0^T (\mathbf{R}_N \boldsymbol{\alpha}) \times & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \delta \mathbf{x}_0 \\ \delta \mathbf{g}_0 \\ \delta \mathbf{q} \end{Bmatrix} \\ & + \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ -\mathbf{f}_N \times & -\mathbf{R}_N \\ \mathbf{I} & \mathbf{0} \\ \left(\mathbf{R}_0 \left(\tilde{\mathbf{f}}_P + \mathbf{U}_P \mathbf{q} \right) \right) \times & \mathbf{R}_N \\ \mathbf{U}_P^T \mathbf{R}_0^T & \mathbf{V}_P^T \mathbf{R}_0^T \mathbf{R}_N \end{bmatrix} \begin{Bmatrix} \delta \boldsymbol{\lambda} \\ \delta \boldsymbol{\alpha} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_N \times \boldsymbol{\lambda} + \mathbf{R}_N \boldsymbol{\alpha} \\ (9.402) \end{Bmatrix} \end{aligned} \quad (9.404)$$

Chapter 10

Beam Element

TODO

10.1 Generalized strains, strain rates and their linearization

Authors: Marco Morandini and Wenguo Zhu

This a draft, but hopefully consistent and correct documentation of the beam strains, strain rates and of their derivatives. Arbitrary point on the beam reference line, and the rotation parameter:

$$p(\xi) = N_i(\xi)(x_i + R_i \tilde{f}_i)$$

$$g(\xi) = N_i(\xi)g_i$$

The generalized strain:

$$\varepsilon(\xi) = R^T(\xi)N'_i(\xi)(x_i + R_i \tilde{f}_i) - \bar{p}'$$

$$\kappa(\xi) = R^T(\xi)(G(\xi)N'_i(\xi)g_i) + \kappa_r$$

Rotation:

$$R = R(N_i g_i) R_r$$

Derivative of Rotation:

$$\dot{R} = \omega \times R$$

$$\dot{R}^T = -R^T \omega \times$$

Section angular velocity:

$$\omega = G(\xi)N'_i(\xi)\dot{g}_i - \omega_r \times N_i g_i$$

And the strain rate can be expressed as:

$$l = N'_i(\xi)(x_i + R_i \tilde{f}_i)$$

$$\begin{aligned}\dot{l} &= N'_{Jk}(\dot{x}_k + \dot{R}_k \tilde{f}_k) \\ &= N'_{Jk}(\dot{x}_k + \omega_k \times R_k \tilde{f}_k)\end{aligned}$$

$$\begin{aligned}\dot{\varepsilon} &= R^T(\dot{l} - \omega \times l) \\ &= R^T N'_{Jk}(\dot{x}_k + \omega_k \times R_k \tilde{f}_k) \\ &\quad - R^T \omega \times N'_i(x_i + R_i \tilde{f}_i)\end{aligned}$$

$$\mathbf{G} = \frac{4}{4 + \mathbf{g}^T \mathbf{g}} (\mathbf{I} + \frac{1}{2} \mathbf{g} \times)$$

$$\dot{G} = -\frac{4}{(4 + \mathbf{g}^T \mathbf{g})^2} (\dot{\mathbf{g}} \otimes \mathbf{g} + \mathbf{g} \otimes \dot{\mathbf{g}}) (\mathbf{I} + \frac{1}{2} \mathbf{g} \times) + \frac{4}{4 + \mathbf{g}^T \mathbf{g}} (\frac{1}{2} \dot{\mathbf{g}} \times)$$

$$\dot{\kappa} = R^T(G(\xi)N'_i(\xi)\dot{g}_i + \dot{G}(\xi)N'_i(\xi)g_i - \omega \times (G(\xi)N'_i(\xi)g_i)) + \dot{\kappa}_r$$

$$\omega = G(N_i g_i) N'_i \dot{g}_i + R(N_i g_i) \omega_r$$

Angular velocity linearization at an evaluation point J :

$$\begin{aligned}\Delta\omega &= \Delta G N_{Ji} \dot{g}_i \\ &\quad + \cancel{\mathcal{O}^{G \approx I} N_{Ji} \Delta \dot{g}_i} \\ &\quad + \Delta R \omega_r \\ &= \frac{1}{2} N_{jk} \Delta g_k \times N'_{Ji} \dot{g}_i \\ &\quad + N_{Ji} \Delta \dot{g}_i \\ &\quad - \omega_r \times N_{Ji} \Delta g_i \\ &= -\frac{1}{2} \cancel{N'_{Ji} \dot{g}_i \times N_{jk} \Delta g_k} \overset{\dot{g} \approx 0}{\cancel{\Delta g_k}} \\ &\quad + N_{Ji} \Delta \dot{g}_i \\ &\quad - \omega_r \times N_{Ji} \Delta g_i \\ &= N_{Ji} \Delta \dot{g}_i - \omega_r \times N_{Ji} \Delta g_i\end{aligned}$$

Rotation linearization at an evaluation point J :

$$\begin{aligned}\Delta R &= \Delta g \times R_r \\ &= (N_{Ji} \Delta g_i) \times R_r\end{aligned}$$

$$\Delta R^T = -R_r^T (N_{Ji} \Delta g_i) \times$$

Rotation rate linearization at an evaluation point J :

$$\begin{aligned}\dot{R} &= \cancel{\dot{R}_\delta R_r} = \cancel{(G_\delta \dot{g})} \times \cancel{R_r} \\ &= \omega \times R\end{aligned}$$

$$\dot{R}^T = -R^T \omega \times$$

$$\begin{aligned}\Delta \dot{R} &= \Delta \omega \times R \\ &\quad + \omega \times \Delta R \\ &= (N_{Ji} \Delta \dot{g}_i - \omega_r \times N_{Ji} \Delta g_i) \times R \\ &\quad + \omega \times N_{Ji} \Delta g_i \times \\ &= N_{Ji} \Delta \dot{g}_i \times R \\ &\quad - N_{Ji} \Delta g_i \otimes \omega_r R \\ &\quad + \omega_r \otimes N_{Ji} R^T \Delta g_i \\ &\quad + \omega \times N_{Ji} \Delta g_i \times R_r\end{aligned}$$

$$\begin{aligned}\Delta \dot{R}_i &= \Delta \dot{g}_i \times R_i \\ &\quad - \Delta g_i \otimes \omega_{ri} R_i \\ &\quad + \omega_{ri} \otimes R_i^T \Delta g_i \\ &\quad + \omega_i \times \Delta g_i \times R_{ri}\end{aligned}$$

$$\begin{aligned}\Delta l &= N'_{Ji} (\Delta x_i + \Delta R_i \tilde{f}_i) \\ &= N'_{Ji} (\Delta x_i - (R_i \tilde{f}_i) \times \Delta g_i)\end{aligned}$$

Strain linearization at an evaluation point J :

$$\begin{aligned}\Delta \varepsilon &= \Delta R^T N'_{Jk} (x_k + R_k \tilde{f}_k) \\ &\quad + R^T N'_{Jk} (\Delta x_k + \Delta R_k \tilde{f}_k) \\ &= R_r^T N'_{Jk} (x_k + R_k \tilde{f}_k) \times (N_{Jm} \Delta g_m) \\ &\quad + R^T N'_{Jk} (\Delta x_k + \Delta g_k \times R_k \tilde{f}_k) \\ &= R_r^T N'_{Jk} (x_k + R_k \tilde{f}_k) \times (N_{Jm} \Delta g_m) \\ &\quad + R_r^T N'_{Jk} (\Delta x_k - R_k \tilde{f}_k \times \Delta g_k)\end{aligned}$$

$$\begin{aligned}\Delta \kappa &= \cancel{\Delta R^T G N'_{Ji} g_i^{g_i \approx 0}} \\ &\quad \cancel{\pm R^T \Delta G N'_{Ji} g_i^{g_i \approx 0}} \\ &\quad + R^T G N'_{Ji} \Delta g_i \\ &= R_r^T N'_{Ji} \Delta g_i\end{aligned}$$

$$\dot{l} = N'_{Ji}(\dot{x}_i + \dot{R}_i \tilde{f}_i)$$

$$\begin{aligned}\Delta\dot{l} &= N'_{Ji}(\Delta\dot{x}_i + \Delta\dot{R}_i \tilde{f}_i) \\ &= N'_{Ji}(\Delta\dot{x}_i + (\Delta\dot{g}_i \times R_i - \Delta g_i \otimes \omega_{ri} R_i + \omega_{ri} \otimes R_i^T \Delta g_i + \omega_i \times \Delta g_i \times R_{ri}) \tilde{f}_i) \\ &= N'_{Ji} \Delta \dot{x}_i \\ &\quad - N'_{Ji}(R_i \tilde{f}_i) \times \Delta \dot{g}_i \\ &\quad - N'_{Ji}(\omega_{ri} R_i \tilde{f}_i) \Delta g_i \\ &\quad + N'_{Ji} \omega_{ri} \otimes \tilde{f}_i R_i^T \Delta g_i \\ &\quad - N'_{Ji} \omega_i \times (R_{ri} \tilde{f}_i) \times \Delta g_i\end{aligned}$$

Strain rate linearization at an evaluation point J :

$$\begin{aligned}\Delta\dot{\varepsilon}_J &= \Delta R^T(\dot{l} - \omega \times l) \\ &\quad + R^T \Delta \dot{l} \\ &\quad - R^T \Delta \omega \times l \\ &\quad - R^T \omega \times \Delta l \\ &= R_r^T(\dot{l} - \omega \times l) \times N_{Ji} \Delta g_i \\ &\quad + R^T N'_{Ji} \Delta \dot{x}_i \\ &\quad - R^T N'_{Ji}(R_i \tilde{f}_i) \times \Delta \dot{g}_i \\ &\quad - R^T N'_{Ji}(\omega_{ri} R_i \tilde{f}_i) \Delta g_i \\ &\quad + R^T N'_{Ji} \omega_{ri} \otimes \tilde{f}_i R_i^T \Delta g_i \\ &\quad - R^T N'_{Ji} \omega_i \times (R_{ri} \tilde{f}_i) \times \Delta g_i \\ &\quad + R^T l \times N_{Ji} \Delta \dot{g}_i \\ &\quad + \underbrace{R^T(\omega_r \times N_{Ji} \Delta g_i)}_{+ (\omega_r \cdot l) R^T N_{Ji} \Delta g_i} \times l \text{ EXPANDED IN THE TWO LINES BELOW} \\ &\quad - R^T \omega_r \otimes l \cdot N_{Ji} \Delta g_i \\ &\quad - R^T \omega \times N'_{Ji} \Delta x_i \\ &\quad + R^T \omega \times N'_{Ji}(R_i \tilde{f}_i) \times \Delta g_i \\ &= R^T N'_{Ji} \Delta \dot{x}_i \\ &\quad - R^T \omega \times N'_{Ji} \Delta x_i \\ &\quad + R_r^T(\dot{l} - \omega \times l) \times N_{Ji} \Delta g_i \\ &\quad - R^T N'_{Ji}(R_i \tilde{f}_i) \times \Delta \dot{g}_i \\ &\quad - R^T N'_{Ji}(\omega_{ri} R_i \tilde{f}_i) \Delta g_i \\ &\quad + R^T N'_{Ji} \omega_{ri} \otimes \tilde{f}_i R_i^T \Delta g_i \\ &\quad - R^T N'_{Ji} \omega_i \times (R_{ri} \tilde{f}_i) \times \Delta g_i \\ &\quad + R^T l \times N_{Ji} \Delta \dot{g}_i \\ &\quad + R^T(\omega_r \cdot l) N_{Ji} \Delta g_i \\ &\quad - R^T \omega_r \otimes l \cdot N_{Ji} \Delta g_i \\ &\quad + R^T \omega \times N'_{Ji}(R_i \tilde{f}_i) \times \Delta g_i\end{aligned}$$

$$\begin{aligned}
\Delta \dot{\kappa}_J(\xi) &= \Delta R^T \left(\cancel{G^{\approx I} N'_{Jk} \dot{g}_k \approx 0} + \cancel{\Delta \dot{G} N'_{Jk} g_k \approx 0} - \omega_J \times (\cancel{G^{\approx I} N'_{Jk} g_k})^{g_k \approx 0} \right) + \dot{\kappa}_r \\
&\quad + R^T (\cancel{\Delta G N'_{Jk} g_k})^{g_k \approx 0} \\
&\quad + R^T (G^{\approx I} N'_{Jk} \Delta \dot{g}_k) \\
&\quad + R^T (\cancel{G^{\approx I} N'_{Ji} g_i})^{g_i \approx 0} \times \Delta \omega \\
&\quad + R^T (\cancel{\Delta G N'_{Ji} g_i}) \times \omega^{g_i \approx 0} \\
&\quad + R^T (G^{\approx I} N'_{Ji} \Delta g_i) \times \omega \\
&= +R^T N'_{Jk} \Delta \dot{g}_k \\
&\quad + R^T (N'_{Ji} \Delta g_i) \times \omega \\
&= +R^T (N'_{Jk} \Delta \dot{g}_k) \\
&\quad - R^T \omega \times (N'_{Ji} \Delta g_i) \\
&= +R^T N'_{Jk} \Delta \dot{g}_k \\
&\quad - R^T \omega \times (N'_{Jm} \Delta g_m)
\end{aligned}$$

10.2 Fully coupled piezoelectric beam

A fully coupled piezoelectric beam, differently from a piezoelectric beam, do contribute to the electric equations of the abstract nodes it is linked to. Each abstract node do represent an electrode. The constitutive law at any given evaluation point is

$$\left\{ \begin{array}{c} \mathbf{F} \\ \mathbf{Q} \end{array} \right\} = \left[\begin{array}{cc} \mathbf{E}_{kk} & \mathbf{E}_{kV} \\ \mathbf{E}_{Vk} & \mathbf{E}_{VV} \end{array} \right] \left\{ \begin{array}{c} \mathbf{k} \\ \mathbf{V} \end{array} \right\} \quad (10.1)$$

where \mathbf{F} are the six beam internal actions at the evaluation point, \mathbf{Q} the vector of electrodes charges per unit of beam length, \mathbf{k} the six beam generalized deformations and \mathbf{V} the electric difference of potential at the electrodes. The equation contribution to the structural nodes are the same of the piezoelectric beam. The difference with respect to the piezoelectric beam is the equations contribution to the abstract nodes. The unknowns of the abstract nodes do represent the electric difference of potential at the electrodes. The equations at the abstract nodes states that the sum of charge flux (the electric current) flowing out from each node must be equal to zero:

$$\sum_i I_i = 0 \quad (10.2)$$

where I_i is the current flowing out from the abstract node due to the connected element i . For this element the current flowing out from the electrodes is

$$\mathbf{I} = - \int_l \dot{Q} ds = - \int_l \mathbf{E}_{Vk} \dot{\mathbf{k}} + \mathbf{E}_{VV} \dot{\mathbf{V}} ds \quad (10.3)$$

The fully coupled beam element thus turns out to be viscoelastic, as its contribution to the electric equations do depend on the time derivative of the beam generalized measure. The integral $\int()ds$ is computed with the same gauss quadrature scheme used for the evaluation of the beam stiffness matrix. The assembled residual contribution is equal to Eq. (10.3) with sign changed.

Chapter 11

Shell Element

Authors: Marco Morandini and Riccardo Vescovini

11.1 Variational principle

The formulation refers to the modified Hu-Washizu variational functional. The linearization of the functional is written as:

$$\delta W + \Delta \delta W = 0 \quad (11.1)$$

with:

$$\delta W = \int_A (\delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} + \delta \hat{\boldsymbol{\epsilon}}^T \boldsymbol{\sigma}) dA \quad (11.2)$$

with

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\epsilon} + \hat{\boldsymbol{\epsilon}}) \quad (11.3)$$

A linear (visco)elastic stress-strain relationship is considered,

$$\boldsymbol{\sigma} = \boldsymbol{C}(\boldsymbol{\epsilon} + \hat{\boldsymbol{\epsilon}}) + \boldsymbol{E}(\dot{\boldsymbol{\epsilon}} + \dot{\hat{\boldsymbol{\epsilon}}}) \quad (11.4)$$

FIXME: $\dot{\hat{\boldsymbol{\epsilon}}}$? Direi di no! and:

$$\Delta \delta W = \int_A (\delta \boldsymbol{\epsilon}^T \boldsymbol{C}(\Delta \boldsymbol{\epsilon} + \Delta \hat{\boldsymbol{\epsilon}}) + \delta \hat{\boldsymbol{\epsilon}}^T \boldsymbol{C}(\Delta \boldsymbol{\epsilon} + \Delta \hat{\boldsymbol{\epsilon}}) + \Delta \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma}) dA \quad (11.5)$$

where:

- \boldsymbol{C} and \boldsymbol{E} are the linear viscoelastic constitutive law matrices (integrated along the shell thickness)
- $\boldsymbol{\epsilon}$ is the vector of the compatible strains (resulting from the strain-displacement relations)
- $\hat{\boldsymbol{\epsilon}}$ is the vector of the enhancing strains (EAS)
- $\boldsymbol{\sigma}$ is the vector of forces and moments per unit length

The enhancing strains $\tilde{\epsilon}$ are interpolated within the element.
The compatible strains ϵ are defined as:

$${}_{12 \times 1} \epsilon = \begin{Bmatrix} \tilde{\epsilon}_1 \\ \tilde{\epsilon}_2 \\ \tilde{\kappa}_1 \\ \tilde{\kappa}_2 \end{Bmatrix} \quad (11.6)$$

where:

$$\tilde{\epsilon}_k = \mathbf{T}^T \mathbf{y}_{,k} - \mathbf{e}_k \quad (11.7)$$

where the comma followed by the index represents the derivation with respect to the k -th arc-length coordinate.

\mathbf{T} is the orientation and \mathbf{y} is the position vector both in the deformed configuration, and $\mathbf{e}_k = \{e_{1k}, e_{2k}, e_{3k}\}$, with $e_{ik} = \delta_{ik}$.

$$\tilde{\kappa}_k = \mathbf{T}^T \boldsymbol{\kappa}_k - \mathbf{T}_0^T \boldsymbol{\kappa}_k^0 \quad (11.8)$$

with:

$$\boldsymbol{\kappa}_k \times = \mathbf{T}_{,k} \mathbf{T}^T \quad (11.9)$$

11.1.1 Strain Rate

The compatible strain derivatives $\dot{\epsilon}$ are defined as:

$${}_{12 \times 1} \dot{\epsilon} = \begin{Bmatrix} \dot{\tilde{\epsilon}}_1 \\ \dot{\tilde{\epsilon}}_2 \\ \dot{\tilde{\kappa}}_1 \\ \dot{\tilde{\kappa}}_2 \end{Bmatrix} \quad (11.10)$$

where:

$$\frac{d}{dt} \tilde{\epsilon}_k = \dot{\mathbf{T}}^T \mathbf{y}_{,k} + \mathbf{T}^T \dot{\mathbf{y}}_{,k} = -\mathbf{T}^T \boldsymbol{\omega} \times \mathbf{y}_{,k} + \mathbf{T}^T \dot{\mathbf{y}}_{,k} \quad (11.11)$$

and

$$\frac{d}{dt} \tilde{\kappa}_k = \dot{\mathbf{T}}^T \boldsymbol{\kappa}_k + \mathbf{T}^T \dot{\boldsymbol{\kappa}}_k = -\mathbf{T}^T \boldsymbol{\omega} \times \boldsymbol{\kappa}_k + \mathbf{T}^T \dot{\boldsymbol{\kappa}}_k. \quad (11.12)$$

Since

$$\frac{d}{dt} (\mathbf{T}_{,k}) = \frac{d}{dt} (\boldsymbol{\kappa}_k \times \mathbf{T}) = \dot{\boldsymbol{\kappa}}_k \times \mathbf{T} + \boldsymbol{\kappa}_k \times \boldsymbol{\omega} \times \mathbf{T} \quad (11.13)$$

$$(\dot{\mathbf{T}})_{,k} = (\boldsymbol{\omega} \times \mathbf{T})_{,k} = \boldsymbol{\omega}_{,k} \times \mathbf{T} + \boldsymbol{\omega} \times \boldsymbol{\kappa}_k \times \mathbf{T} \quad (11.14)$$

then

$$\dot{\boldsymbol{\kappa}}_k \times + \boldsymbol{\kappa}_k \times \boldsymbol{\omega} \times = \boldsymbol{\omega}_{,k} \times + \boldsymbol{\omega} \times \boldsymbol{\kappa}_k \times \quad (11.15)$$

i.e.

$$\dot{\boldsymbol{\kappa}}_k = \boldsymbol{\omega}_{,k} + \boldsymbol{\omega} \times \boldsymbol{\kappa}_k \quad (11.16)$$

Thus

$$\frac{d}{dt} \tilde{\kappa}_k = \mathbf{T}^T \boldsymbol{\omega}_{,k} \quad (11.17)$$

The vector $\boldsymbol{\sigma}$ is:

$$\boldsymbol{\sigma} = \begin{Bmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \\ \mathbf{m}_1 \\ \mathbf{m}_2 \end{Bmatrix} \quad (11.18)$$

11.2 Finite element discretization and notation

The shell elements has four nodes, numbered counterclockwise starting from the upper right node, as shown in Figure 11.1. In the natural domain, the coordinate system is identified by the coordinates ξ_1

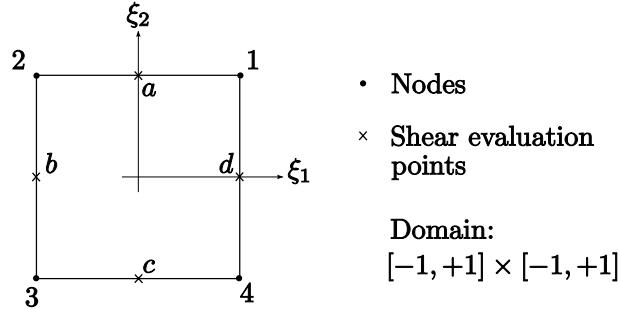


Figure 11.1: Shell finite element

and ξ_2 , both varying in the range $[-1, 1]$.

In the following, the notation

$$\boldsymbol{\xi} = (\xi_1, \xi_2) \quad (11.19)$$

will be used. Four points are defined at the element's mid-sides. They are used in the context of the ANS approach. In particular, they are denoted by the set of coordinates:

$$\begin{aligned} \boldsymbol{\xi}_a &= (\xi_{1a}, \xi_{2a}) \\ \boldsymbol{\xi}_b &= (\xi_{1b}, \xi_{2b}) \\ \boldsymbol{\xi}_c &= (\xi_{1c}, \xi_{2c}) \\ \boldsymbol{\xi}_d &= (\xi_{1d}, \xi_{2d}) \end{aligned} \quad (11.20)$$

The following convention is adopted:

- the subscript n denotes a nodal variable
- $\boldsymbol{\xi}_i$ means $\boldsymbol{\xi}$ evaluated in a generic integration point
- $\boldsymbol{\xi}_A$ means $\boldsymbol{\xi}$ evaluated in one of the four shear evaluation points
- $\boldsymbol{\xi}_0$ means $\boldsymbol{\xi}$ evaluated at the origin

According to the numbering of the nodes, the bilinear shape functions defined at element level are:

$$L_1(\xi) = \frac{1}{4} (1 + \xi_1) (1 + \xi_2) \quad (11.21a)$$

$$L_2(\xi) = \frac{1}{4} (1 - \xi_1) (1 + \xi_2) \quad (11.21b)$$

$$L_3(\xi) = \frac{1}{4} (1 - \xi_1) (1 - \xi_2) \quad (11.21c)$$

$$L_4(\xi) = \frac{1}{4} (1 + \xi_1) (1 - \xi_2) \quad (11.21d)$$

The degrees of freedom associated to each element are 24 nodal position and orientation parameter components, collected in vector \mathbf{q} :

$$\delta_{24 \times 1} \mathbf{q} = \left\{ \begin{array}{l} \delta \mathbf{y}_n \\ \varphi_{\delta n} \end{array} \right\} = \left\{ \begin{array}{l} \delta \mathbf{y}_1 \\ \varphi_{\delta 1} \\ \delta \mathbf{y}_2 \\ \varphi_{\delta 2} \\ \delta \mathbf{y}_3 \\ \varphi_{\delta 3} \\ \delta \mathbf{y}_4 \\ \varphi_{\delta 4} \end{array} \right\} \quad \Delta_{24 \times 1} \mathbf{q} = \left\{ \begin{array}{l} \Delta \mathbf{y}_1 \\ \varphi_{\Delta 1} \\ \Delta \mathbf{y}_2 \\ \varphi_{\Delta 2} \\ \Delta \mathbf{y}_3 \\ \varphi_{\Delta 3} \\ \Delta \mathbf{y}_4 \\ \varphi_{\Delta 4} \end{array} \right\} \quad (11.22)$$

and 7 internal degrees of freedom related to the assumed strain (EAS) and collected in vector β . The total number of degrees of freedom is 31.

11.3 Linearization

The virtual variation of the compatible strains ϵ is

$$\delta \epsilon = \left\{ \begin{array}{l} \delta \tilde{\epsilon}_1 \\ \delta \tilde{\epsilon}_2 \\ \delta \tilde{\kappa}_1 \\ \delta \tilde{\kappa}_2 \end{array} \right\} \quad (11.23)$$

In particular, the virtual variations of the linear deformation $\tilde{\epsilon}_k$ is:

$$\delta \tilde{\epsilon}_k = \delta (\mathbf{T}^T \mathbf{y}_{,k}) = \sum_{i=1}^n \mathbf{T}^T \mathbf{y}_{,k} \times \Phi_n L_n \varphi_{\delta n} + \sum_{i=1}^n \mathbf{T}^T L_{n,k} \delta \mathbf{y}_n \quad (11.24)$$

where:

$$\Phi_n = \bar{\mathbf{T}} \tilde{\Gamma}_n^{-1} \bar{\mathbf{T}}^T \quad (11.25)$$

The virtual variations of the curvatures $\tilde{\kappa}_k$ is:

$$\delta \tilde{\kappa}_k = \delta (\mathbf{T}^T \kappa_k) = \sum_{i=1}^n \mathbf{T}^T \kappa_k \times \Phi_n L_n \varphi_{\delta n} + \sum_{i=1}^n \mathbf{T}^T K_{kn} \varphi_{\delta n} \quad (11.26)$$

where:

$$K_{kn} = \bar{\mathbf{T}} \mathcal{L}(\tilde{\varphi}, \tilde{\varphi}_{,k}) \tilde{\Gamma}_n^{-1} \bar{\mathbf{T}}^T L_n + \Phi_n L_{n,k} \quad (11.27)$$

the term $L_{n,k}$ is the derivative of the n -th shape function with respect to the k -th arc-length coordinate. The last term of Eq. 11.5 requires the evaluation of the terms:

$$\begin{aligned} \mathbf{n}_k \cdot \Delta\delta\tilde{\epsilon}_k = \mathbf{n}_k \cdot \Delta\delta(\mathbf{T}^T \mathbf{y}_{,k}) &= \sum_{m,n=1}^4 \boldsymbol{\varphi}_{\delta n} \boldsymbol{\Phi}_n^T L_n \mathbf{y}_{,k} \times (\mathbf{T} \mathbf{n}_k) \times \boldsymbol{\Phi}_m L_m \boldsymbol{\varphi}_{\Delta m} + \\ &+ \sum_{m,n=1}^4 \boldsymbol{\varphi}_{\delta n} \boldsymbol{\Phi}_n^T L_n (\mathbf{T} \mathbf{n}_k) \times L_{m,k} \Delta \mathbf{y}_n + \\ &- \sum_{m,n=1}^4 \delta \mathbf{y}_n L_{n,k} (\mathbf{T} \mathbf{n}_k) \times \boldsymbol{\Phi}_m L_m \boldsymbol{\varphi}_{\Delta m} \end{aligned} \quad (11.28)$$

$$\begin{aligned} \mathbf{m}_k \cdot \Delta\delta\tilde{\kappa}_k = \mathbf{m}_k \cdot \Delta\delta(\mathbf{T}^T \boldsymbol{\kappa}_k) &= \sum_{m,n=1}^4 \boldsymbol{\varphi}_{\delta n} L_n \boldsymbol{\Phi}_n^T \boldsymbol{\kappa}_k \times (\mathbf{T} \mathbf{m}_k) \times \boldsymbol{\Phi}_m L_m \boldsymbol{\varphi}_{\Delta m} + \\ &+ \sum_{m,n=1}^4 \boldsymbol{\varphi}_{\delta n} L_n \boldsymbol{\Phi}_n^T (\mathbf{T} \mathbf{m}_k) \times \boldsymbol{\Phi}_m L_{m,k} \boldsymbol{\varphi}_{\Delta m} \end{aligned} \quad (11.29)$$

11.4 Structural Damping

Consistently formulated structural damping requires to express the time derivatives of strain and curvature. A linear contribution to internal force and moment fluxes is considered,

$$\boldsymbol{\sigma} += \mathbf{E}\dot{\boldsymbol{\epsilon}} \quad (11.30)$$

The discretized strain rate is

$$\frac{d}{dt} \tilde{\boldsymbol{\epsilon}}_k = \mathbf{T}^T \mathbf{y}_{,k} \times \sum_{i=1}^n \boldsymbol{\Phi}_i L_i \boldsymbol{\omega}_i + \mathbf{T}^T \sum_{i=1}^n L_{n,k} \dot{\mathbf{y}}_i \quad (11.31)$$

The discretized curvature rate is

$$\frac{d}{dt} \tilde{\boldsymbol{\kappa}}_k = \mathbf{T}^T \sum_{i=1}^n L_{n,k} \boldsymbol{\omega}_i \quad (11.32)$$

11.5 Implementation

Having shown the variational principle and the linearization of strains and curvatures, it is then possible to develop the procedure to derive the Jacobian matrix and the residual.

11.5.1 Orientation interpolation

In the initial configuration it is built the matrix:

$$iT\mathbf{a}_n = \mathbf{R}_n^T [\mathbf{t}_{1n} \mathbf{t}_{2n} \mathbf{t}_{3n}] \quad (11.33)$$

with \mathbf{R}_n node orientation, and \mathbf{t}_{in} vectors tangent to the element surface.

So:

$$\mathbf{T}_n = \mathbf{R}_n iT\mathbf{a}_n \quad (11.34)$$

It is then calculated:

$$\mathbf{T}_{\text{avg}} = \frac{1}{4} \sum_{n=1}^4 \mathbf{T}_n \quad (11.35)$$

which is in general a not orthogonal matrix.

An orthogonal matrix can be obtained by applying:

$$\bar{\mathbf{T}} = \text{Rot}(\text{VecRot}(\mathbf{T}_{\text{avg}})) \quad (11.36)$$

which is then used to calculate the nodal rotation:

$$\tilde{\mathbf{R}}_n = \bar{\mathbf{T}}^T \mathbf{T}_n \quad (11.37)$$

and:

$$\tilde{\varphi}_n = \text{VecRot}(\tilde{\mathbf{R}}_n) \quad (11.38)$$

The nodal values of $\tilde{\varphi}_n$ are interpolated by means of the shape functions:

$$\tilde{\varphi}(\xi_i) = \sum_{i=1}^4 L_n(\xi_i) \tilde{\varphi}_n \quad (11.39)$$

from which is obtained the rotation matrix at the integration point:

$$\tilde{\mathbf{R}}(\xi_i) = \text{Rot}(\tilde{\varphi}(\xi_i)) \quad (11.40)$$

and finally the orientation at the integration point:

$$\mathbf{T}(\xi_i) = \bar{\mathbf{T}} \tilde{\mathbf{R}}(\xi_i) \quad (11.41)$$

The virtual variation of φ at the integration point is related to the nodal values by:

$$\varphi_\delta(\xi_i) = \sum_{n=1}^4 \Phi_n(\xi_i) L_n(\xi_i) \varphi_{\delta n} \quad (11.42)$$

with:

$$\Phi_n(\xi_i) = \bar{\mathbf{T}} \tilde{\Gamma}(\tilde{\varphi}(\xi_i)) \tilde{\Gamma}^{-1}(\tilde{\varphi}_n) \bar{\mathbf{T}}^T \quad (11.43)$$

Similarly the virtual variation of φ can be obtained at the shear evaluation points as:

$$\varphi_\delta(\xi_A) = \sum_{n=1}^4 \Phi_n(\xi_A) L_n(\xi_A) \varphi_{\delta n} \quad (11.44)$$

11.5.2 Position interpolation

The nodal positions are so interpolated as:

$$\mathbf{y} = \sum_{n=1}^4 L_n \mathbf{y}_n \quad (11.45)$$

The derivatives with respect to the generic coordinate ξ_k at the integration points is:

$$\frac{\partial \mathbf{x}(\xi_i)}{\partial \xi_k} = \sum_{n=1}^4 \frac{\partial L_n(\xi_i)}{\partial \xi_k} \mathbf{x}_n \quad (11.46)$$

It is then possible to build the matrix:

$$\mathbf{S}_{\alpha\beta}(\xi_i) = \begin{bmatrix} t_1^0(\xi_i) \frac{\partial \mathbf{x}(\xi_i)}{\partial \xi_1} & t_1^0(\xi_i) \frac{\partial \mathbf{x}(\xi_i)}{\partial \xi_2} \\ t_2^0(\xi_i) \frac{\partial \mathbf{x}(\xi_i)}{\partial \xi_1} & t_2^0(\xi_i) \frac{\partial \mathbf{x}(\xi_i)}{\partial \xi_2} \end{bmatrix} \quad (11.47)$$

observing that $\mathbf{T}_0(\xi_i) = [t_1^0(\xi_i) \ t_2^0(\xi_i) \ t_3^0(\xi_i)]$.

With the same procedure are derived $\mathbf{S}_{\alpha\beta}(\xi_0)$ and $\mathbf{S}_{\alpha\beta}(\xi_A)$.

$$\mathbf{L}_{\alpha B}(\xi_i) = \begin{bmatrix} \frac{\partial L_1(\xi_i)}{\partial \xi_1} & \frac{\partial L_1(\xi_i)}{\partial \xi_2} \\ \frac{\partial L_2(\xi_i)}{\partial \xi_1} & \frac{\partial L_2(\xi_i)}{\partial \xi_2} \\ \frac{\partial L_3(\xi_i)}{\partial \xi_1} & \frac{\partial L_3(\xi_i)}{\partial \xi_2} \\ \frac{\partial L_4(\xi_i)}{\partial \xi_1} & \frac{\partial L_4(\xi_i)}{\partial \xi_2} \end{bmatrix} \quad (11.48)$$

$$\mathbf{L}_{\alpha\beta}(\xi_i) = \mathbf{L}_{\alpha B}(\xi_i) \mathbf{S}_{\alpha\beta}^{-1}(\xi_i) \quad (11.49)$$

which is the matrix giving the variation of the shape functions L_n with respect the the arc-length coordinates:

$$\mathbf{L}_{\alpha\beta} = \begin{bmatrix} L_{1,1} & L_{1,2} \\ L_{2,1} & L_{2,2} \\ L_{3,1} & L_{3,2} \\ L_{4,1} & L_{4,2} \end{bmatrix} \quad (11.50)$$

In order to perform the integration in the isoparametric domain the area element need to be calculated. In particular, the determinant of the Jacobian of the transformation between the physical and the natural domain has to be calculated. It is:

$$\alpha(\xi_i) = \det \mathbf{S}_{\alpha\beta}(\xi_i) \quad \alpha(\xi_A) = \det \mathbf{S}_{\alpha\beta}(\xi_A) \quad (11.51)$$

11.5.3 Enhancing strains interpolation

The virtual variation and increments of the enhancing strains $\hat{\epsilon}$ are interpolated as:

$$\delta \hat{\epsilon}(\xi_i) = \mathbf{P}_{12 \times 7}(\xi_i) \delta \boldsymbol{\beta} \quad \Delta \hat{\epsilon}(\xi_i) = \mathbf{P}(\xi_i) \Delta \boldsymbol{\beta} \quad (11.52)$$

where $\boldsymbol{\beta}$ is the vector collecting the the strains parameters, while the expression of the matrix \mathbf{P} is derived in the next.

The EAS approach here adopted considers the enhancing of the membrane strains only, i.e. $\epsilon_{11}, \epsilon_{12}, \epsilon_{21}$

and ϵ_{22} . Shear strains and bending strains are not enhanced.

The interpolation is performed in the natural domain adopting the shape functions:

$$\mathbf{H}(\boldsymbol{\xi}_i) = \begin{bmatrix} \xi_{1i} & \xi_{1i}\xi_{2i} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \xi_{2i} & \xi_{1i}\xi_{2i} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \xi_{1i} & \xi_{1i}\xi_{2i} & 0 \\ 0 & 0 & 0 & 0 & 0 & \xi_{1i}\xi_{2i} & \xi_{2i} \end{bmatrix} \quad (11.53)$$

The interpolation is reported in the physical domain with a push-forward operation with the transformation matrix \mathbf{M}_0 , defined as:

$$\mathbf{M}_0 = \begin{bmatrix} s_{1,1}s_{1,1} & s_{1,2}s_{1,2} & s_{1,2}s_{1,1} & s_{1,1}s_{1,2} \\ s_{2,1}s_{2,1} & s_{2,2}s_{2,2} & s_{2,2}s_{2,1} & s_{2,1}s_{2,2} \\ s_{1,1}s_{2,1} & s_{1,2}s_{2,2} & s_{1,1}s_{2,2} & s_{1,2}s_{2,1} \\ s_{1,1}s_{2,1} & s_{1,2}s_{2,2} & s_{1,2}s_{2,1} & s_{1,1}s_{2,2} \end{bmatrix} \quad (11.54)$$

where the generic term $s_{i,k}$ is the element (i, k) of the matrix $\mathbf{S}_{\alpha\beta}(\boldsymbol{\xi}_0)$.

$$\mathbf{P}(\boldsymbol{\xi}_i) = \frac{\alpha(\boldsymbol{\xi}_0)}{\alpha(\boldsymbol{\xi}_i)} \mathbf{P} \mathbf{M}_0^{-T} \mathbf{H}(\boldsymbol{\xi}_i) \quad (11.55)$$

where \mathbf{P} is a permutation matrix defined as:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (11.56)$$

which has the places the enhanced strains in the corresponding positions of the global vector of membrane and bending strains $\boldsymbol{\epsilon}$.

11.5.4 Compatible strains interpolation

The virtual variations of the linear deformation $\tilde{\boldsymbol{\epsilon}}_k$ at the i -th integration point is obtained from Eq. 11.24:

$$\delta\tilde{\boldsymbol{\epsilon}}_k(\boldsymbol{\xi}_i) = \delta \left(\mathbf{T}(\boldsymbol{\xi}_i)^T \mathbf{y}_{,k}(\boldsymbol{\xi}_i) \right) = \sum_{i=1}^n \mathbf{T}(\boldsymbol{\xi}_i)^T \mathbf{y}_{,k}(\boldsymbol{\xi}_i) \times \Phi_n(\boldsymbol{\xi}_i) L_n(\boldsymbol{\xi}_i) \varphi_{n\delta} + \sum_{i=1}^n \mathbf{T}(\boldsymbol{\xi}_i)^T \mathbf{L}_{\alpha\beta}(\boldsymbol{\xi}_i) (n, k)(\boldsymbol{\xi}_i) \delta \mathbf{y}_n \quad (11.57)$$

where $\mathbf{L}_{\alpha\beta}(\xi_i)(n, k)$ denotes the element (n, k) of the matrix $\mathbf{L}_{\alpha\beta}(\xi_i)$.

The virtual variation of the curvatures $\tilde{\kappa}_k$ at the i -th integration point is derived from Eq. 11.26:

$$\begin{aligned}\delta \tilde{\kappa}_k(\xi_i) &= \delta \left(\mathbf{T}(\xi_i)^T \boldsymbol{\kappa}_k(\xi_i) \right) = \\ &\sum_{i=1}^n \mathbf{T}(\xi_i)^T \boldsymbol{\kappa}_k(\xi_i) \times \Phi_n(\xi_i) L_n(\xi_i) \varphi_{n\delta} + \sum_{i=1}^n \mathbf{T}(\xi_i)^T \mathbf{K}_{kn}(\xi_i) \varphi_{n\delta}\end{aligned}\quad (11.58)$$

where:

$$\Phi_n(\xi_i) = \bar{\mathbf{T}} \tilde{\Gamma}(\tilde{\varphi}(\xi_i)) \tilde{\Gamma}_n^{-1} \bar{\mathbf{T}}^T \quad (11.59)$$

and:

$$\mathbf{K}_{kn}(\xi_i) = \bar{\mathbf{T}} \mathcal{L}(\tilde{\varphi}(\xi_i), \tilde{\varphi}_{,k}(\xi_i)) \tilde{\Gamma}_n^{-1} \bar{\mathbf{T}}^T L_n(\xi_i) + \Phi_n(\xi_i) \mathbf{L}_{\alpha\beta}(\xi_i)(n, k)(\xi_i) \quad (11.60)$$

The derivative of the vector $\tilde{\varphi}$ with respect to the k -th arc-length coordinate can be expressed as function of the nodal values as:

$$\tilde{\varphi}_{,k}(\xi_i) = \sum_{n=1}^4 \mathbf{L}_{\alpha\beta}(\xi_i)(n, k) \varphi_n \quad (11.61)$$

The curvatures are given by:

$$\boldsymbol{\kappa}_k(\xi_i) = \bar{\mathbf{T}} \tilde{\Gamma}(\tilde{\varphi}(\xi_i)) \tilde{\varphi}_{,k}(\xi_i) \quad (11.62)$$

and the derivative of the position vector is:

$$\mathbf{y}_{,k}(\xi_i) = \sum_{i=1}^4 \mathbf{L}_{\alpha\beta}(\xi_i)(n, k) \mathbf{y}_n \quad (11.63)$$

The compatible strains can so be expressed as function of the nodal variables \mathbf{q} by considering Eqs. 11.57 and 11.58:

$$\delta \boldsymbol{\epsilon} = \bar{\mathbf{B}} \delta \mathbf{q} \quad (11.64)$$

with:

$$\bar{\mathbf{B}}(\xi_i) = \begin{bmatrix} \bar{\mathbf{B}}_1(\xi_i) & \bar{\mathbf{B}}_2(\xi_i) & \bar{\mathbf{B}}_3(\xi_i) & \bar{\mathbf{B}}_4(\xi_i) \end{bmatrix}_{12 \times 24} \quad (11.65)$$

and:

$$\bar{\mathbf{B}}_n(\xi_i) = \begin{bmatrix} \mathbf{T}(\xi_i)^T \mathbf{L}_{\alpha\beta}(\xi_i)(n, 1) & \mathbf{T}(\xi_i)^T \mathbf{y}_{,1}(\xi_i) \times \Phi_n(\xi_i) L_n(\xi_i) \\ \mathbf{T}(\xi_i)^T \mathbf{L}_{\alpha\beta}(\xi_i)(n, 2) & \mathbf{T}(\xi_i)^T \mathbf{y}_{,2}(\xi_i) \times \Phi_n(\xi_i) L_n(\xi_i) \\ 0 & \mathbf{T}(\xi_i)^T \boldsymbol{\kappa}_1(\xi_i) \times \Phi_n(\xi_i) L_n(\xi_i) + \mathbf{T}(\xi_i)^T \mathbf{K}_{1n}(\xi_i) \\ 0 & \mathbf{T}(\xi_i)^T \boldsymbol{\kappa}_2(\xi_i) \times \Phi_n(\xi_i) L_n(\xi_i) + \mathbf{T}(\xi_i)^T \mathbf{K}_{2n}(\xi_i) \end{bmatrix}_{12 \times 6} \quad (11.66)$$

11.5.5 ANS

The ANS technique is applied to prevent shear locking. Here the ANS is applied to shear strains only, i.e. ϵ_1 and ϵ_2 . The approach consists in interpolating the shear strains in the mid-sides of the element in the natural domain. Such interpolated strains are then used to derive the shear strains at the integration points.

The use of the ANS technique leads a re-definition of:

- the two rows of the $\bar{\mathbf{B}}$ matrix relative to shear strains
- the shear strains

The compatible strains are:

$$\tilde{\boldsymbol{\epsilon}}_1 = \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_1 \end{Bmatrix} \quad \tilde{\boldsymbol{\epsilon}}_2 = \begin{Bmatrix} \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_2 \end{Bmatrix} \quad (11.67)$$

At first, the compatible strains are evaluated at the shear evaluation points of Figure 11.1. This is done referring to Eq. 11.7

$$\tilde{\boldsymbol{\epsilon}}_k(\xi_A) = \mathbf{T}(\xi_A)^T \mathbf{y}_{,k}(\xi_A) - \mathbf{e}_k \quad (11.68)$$

The matrix $\bar{\mathbf{B}}$ of Eq. 11.65 is built also in the shear evaluation points for the membrane strains only, and is indicated as $\bar{\bar{\mathbf{B}}}$

$$\bar{\bar{\mathbf{B}}} \underset{6 \times 24}{=} \left[\begin{array}{cccc} \bar{\bar{\mathbf{B}}}_1(\xi_A) & \bar{\bar{\mathbf{B}}}_2(\xi_A) & \bar{\bar{\mathbf{B}}}_3(\xi_A) & \bar{\bar{\mathbf{B}}}_4(\xi_A) \end{array} \right] \quad (11.69)$$

with:

$$\bar{\bar{\mathbf{B}}}_n \underset{6 \times 6}{=} \left[\begin{array}{cc} \mathbf{T}(\xi_A)^T \mathbf{L}_{\alpha\beta}(\xi_A)(n,1) & \mathbf{T}(\xi_A)^T \mathbf{y}_{,1}(\xi_A) \times \Phi_n(\xi_A) L_n(\xi_A) \\ \mathbf{T}(\xi_A)^T \mathbf{L}_{\alpha\beta}(\xi_A)(n,2) & \mathbf{T}(\xi_A)^T \mathbf{y}_{,2}(\xi_A) \times \Phi_n(\xi_A) L_n(\xi_A) \end{array} \right] \quad (11.70)$$

The rows relative to the shear strains ϵ_1 and ϵ_2 are collected in:

$$\bar{\bar{\mathbf{B}}}\bar{\mathbf{3}} \underset{4 \times 24}{=} \left[\begin{array}{c} \bar{\bar{\mathbf{B}}}(\xi_a)(3,:) \\ \bar{\bar{\mathbf{B}}}(\xi_b)(3,:) \\ \bar{\bar{\mathbf{B}}}(\xi_c)(3,:) \\ \bar{\bar{\mathbf{B}}}(\xi_d)(3,:) \end{array} \right] \quad \bar{\bar{\mathbf{B}}}\bar{\mathbf{6}} \underset{4 \times 24}{=} \left[\begin{array}{c} \bar{\bar{\mathbf{B}}}(\xi_a)(6,:) \\ \bar{\bar{\mathbf{B}}}(\xi_b)(6,:) \\ \bar{\bar{\mathbf{B}}}(\xi_c)(6,:) \\ \bar{\bar{\mathbf{B}}}(\xi_d)(6,:) \end{array} \right] \quad (11.71)$$

where $\bar{\bar{\mathbf{B}}}(\xi_a)(k,:)$ is used to denote all the element of the k -th row of $\bar{\bar{\mathbf{B}}}(\xi_a)$.

The matrices $\bar{\bar{\mathbf{B}}}\bar{\mathbf{3}}$ and $\bar{\bar{\mathbf{B}}}\bar{\mathbf{6}}$ are interpolated to obtain the values at the integration points.

$$\bar{\bar{\mathbf{B}}}\bar{\mathbf{3}}(\xi_i) = \frac{1}{2} \left[\begin{array}{cccc} 1 + \xi_{2i} & 0 & 1 - \xi_{2i} & 0 \end{array} \right] \bar{\bar{\mathbf{B}}}\bar{\mathbf{3}}(\xi_A) \quad (11.72)$$

$$\bar{\bar{\mathbf{B}}}\bar{\mathbf{6}}(\xi_i) = \frac{1}{2} \left[\begin{array}{cccc} 1 + \xi_{1i} & 0 & 1 - \xi_{1i} & 0 \end{array} \right] \bar{\bar{\mathbf{B}}}\bar{\mathbf{6}}(\xi_A) \quad (11.73)$$

The resulting matrices $\bar{\bar{\mathbf{B}}}\bar{\mathbf{3}}$ and $\bar{\bar{\mathbf{B}}}\bar{\mathbf{6}}$ replace the corresponding rows of the matrix $\bar{\mathbf{B}}$ of Eq. 11.65.

The second step in the application of the ANS technique is the interpolation of the shear strains:

$$\epsilon_1(\xi_i) = \frac{1}{2} (1 + \xi_{2i}) \epsilon_1(\xi_a) + \frac{1}{2} (1 - \xi_{2i}) \epsilon_1(\xi_c) \quad (11.74a)$$

$$\epsilon_2(\xi_i) = \frac{1}{2} (1 - \xi_{1i}) \epsilon_2(\xi_b) + \frac{1}{2} (1 + \xi_{1i}) \epsilon_2(\xi_d) \quad (11.74b)$$

The so interpolated strains ϵ_1 and ϵ_2 are then inserted into the proper position in the vector $\boldsymbol{\epsilon}$ respectively.

11.5.6 Forces

Strains and curvatures can be calculated as:

$$\tilde{\epsilon}_k(\xi_i) = \mathbf{T}(\xi_i)^T \mathbf{y}_{,k}(\xi_i) - \mathbf{e}_k \quad (11.75)$$

$$\tilde{\kappa}_k = \mathbf{T}(\xi_i)^T \boldsymbol{\kappa}_k(\xi_i) - \mathbf{T}_0^T(\xi_i) \boldsymbol{\kappa}_k^0(\xi_i) \quad (11.76)$$

where \mathbf{T}_0 and $\boldsymbol{\kappa}_k^0$ are calculated at the first step.

The total strain is the sum of the compatible and the enhancing strains:

$$\boldsymbol{\epsilon}_t(\xi_i) = \boldsymbol{\epsilon}(\xi_i) + \hat{\boldsymbol{\epsilon}}(\xi_i) \quad (11.77)$$

The resulting forces are easily computed by means of the constitutive law:

$$\boldsymbol{\sigma}(\xi_i) = \mathbf{C}(\xi_i) \boldsymbol{\epsilon}_t(\xi_i) \quad (11.78)$$

which is:

$$\boldsymbol{\sigma}(\xi_i) = \begin{Bmatrix} \mathbf{n}_1(\xi_i) \\ \mathbf{n}_2(\xi_i) \\ \mathbf{m}_1(\xi_i) \\ \mathbf{m}_2(\xi_i) \end{Bmatrix} \quad (11.79)$$

11.5.7 Jacobian matrix

The second variation of the variational principle of Eq. 11.5 gives the Jacobian matrix. In particular, the finite element discretization is here presented for the five different terms of Eq. 11.5.

The first term is:

$$\begin{aligned} \int_A \delta \boldsymbol{\epsilon}^T \mathbf{C} \Delta \boldsymbol{\epsilon} dA &= \\ &= \delta \mathbf{q}^T \sum_{i=1}^4 \overline{\mathbf{B}}(\xi_i)^T \mathbf{C}(\xi_i) \overline{\mathbf{B}}(\xi_i) \alpha(\xi_i) w_i \Delta \mathbf{q} \\ &= \delta \mathbf{q}^T \mathbf{K}_m \Delta \mathbf{q} \end{aligned} \quad (11.80)$$

The second term is:

$$\begin{aligned} \int_A \delta \hat{\boldsymbol{\epsilon}}^T \mathbf{C} \Delta \boldsymbol{\epsilon} dA &= \\ &= \delta \boldsymbol{\beta}^T \sum_{i=1}^4 \mathbf{P}(\xi_i)^T \mathbf{C}(\xi_i) \overline{\mathbf{B}}(\xi_i) \alpha(\xi_i) w_i \Delta \mathbf{q} \\ &= \delta \boldsymbol{\beta}^T \mathbf{K}_{\beta q} \Delta \mathbf{q} \end{aligned} \quad (11.81)$$

The third term is:

$$\begin{aligned} \int_A \delta \boldsymbol{\epsilon}^T \mathbf{C} \Delta \hat{\boldsymbol{\epsilon}} dA &= \\ &= \delta \mathbf{q}^T \mathbf{K}_{\beta q}^T \Delta \boldsymbol{\beta} \end{aligned} \quad (11.82)$$

The fourth term is:

$$\begin{aligned}
\int_A \delta \hat{\epsilon}^T \mathbf{C} \Delta \hat{\epsilon} dA &= \\
&= \delta \boldsymbol{\beta}^T \sum_{i=1}^4 \mathbf{P}(\xi_i)^T \mathbf{C}(\xi_i) \mathbf{P}(\xi_i) \alpha(\xi_i) w_i \Delta \boldsymbol{\beta} \\
&= \delta \boldsymbol{\beta}^T \mathbf{K}_{\beta \beta} \Delta \boldsymbol{\beta}
\end{aligned} \tag{11.83}$$

The fifth term is:

$$\begin{aligned}
\int_A \Delta \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} dA &= \\
&= \delta \mathbf{q}^T \sum_{i=1}^4 \overline{\mathbf{D}}(\xi_i)^T \mathbf{G}(\xi_i) \overline{\mathbf{D}}(\xi_i) \alpha(\xi_i) w_i \Delta \mathbf{q} \\
&= \delta \mathbf{q}^T \mathbf{K}_g \Delta \mathbf{q}
\end{aligned} \tag{11.84}$$

where this last expression is obtained starting from Eqs. 11.28 and 11.29, and where:

$$\overline{\mathbf{D}}(\xi_i) = \begin{bmatrix} \overline{\mathbf{D}}_1(\xi_i) & \overline{\mathbf{D}}_2(\xi_i) & \overline{\mathbf{D}}_3(\xi_i) & \overline{\mathbf{D}}_4(\xi_i) \end{bmatrix}_{15 \times 24} \tag{11.85}$$

with:

$$\overline{\mathbf{D}}_n(\xi_i) = \begin{bmatrix} \mathbf{L}_{\alpha \beta}(\xi_i)(n, 1) \mathbf{I} & \mathbf{0} \\ \mathbf{L}_{\alpha \beta}(\xi_i)(n, 2) \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{1n}(\xi_i) \\ \mathbf{0} & \mathbf{K}_{2n}(\xi_i) \\ \mathbf{0} & \Phi_n(\xi_i) L_n(\xi_i) \end{bmatrix}_{15 \times 6} \tag{11.86}$$

$$\begin{aligned}
\mathbf{Hh}(\xi_i) &= \mathbf{T}(\xi_i) \mathbf{n}_1(\xi_i) \otimes \mathbf{y}_{,1}(\xi_i) - \mathbf{T}(\xi_i) \mathbf{n}_1 \cdot \mathbf{y}_{,1}(\xi_i) \mathbf{I} + \\
&+ \mathbf{T}(\xi_i) \mathbf{n}_2(\xi_i) \otimes \mathbf{y}_{,2}(\xi_i) - \mathbf{T}(\xi_i) \mathbf{n}_2 \cdot \mathbf{y}_{,2}(\xi_i) \mathbf{I} + \\
&+ \mathbf{T}(\xi_i) \mathbf{m}_1(\xi_i) \otimes \boldsymbol{\kappa}_1(\xi_i) - \mathbf{T}(\xi_i) \mathbf{m}_1 \cdot \boldsymbol{\kappa}_1(\xi_i) \mathbf{I} + \\
&+ \mathbf{T}(\xi_i) \mathbf{m}_2(\xi_i) \otimes \boldsymbol{\kappa}_2(\xi_i) - \mathbf{T}(\xi_i) \mathbf{m}_2 \cdot \boldsymbol{\kappa}_2(\xi_i) \mathbf{I}
\end{aligned} \tag{11.87}$$

$$\mathbf{G}(\xi_i) = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{T}(\xi_i) \mathbf{n}_1(\xi_i) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{T}(\xi_i) \mathbf{n}_2(\xi_i) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{T}(\xi_i) \mathbf{n}_1(\xi_i) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}(\xi_i) \mathbf{n}_1(\xi_i) & \mathbf{T}(\xi_i) \mathbf{n}_2(\xi_i) & \mathbf{T}(\xi_i) \mathbf{m}_1(\xi_i) & \mathbf{T}(\xi_i) \mathbf{m}_2(\xi_i) & \mathbf{Hh}(\xi_i) \end{bmatrix}_{15 \times 15} \tag{11.88}$$

11.5.8 Residual

The residual comes from the finite element approximation of Eq. 11.2.

The first term is:

$$\begin{aligned}
\int_A \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} dA &= \\
&= \delta \mathbf{q}^T \sum_{i=1}^4 \overline{\mathbf{B}}(\xi_i)^T \boldsymbol{\sigma}(\xi_i) \alpha(\xi_i) w_i \\
&= \delta \mathbf{q}^T \mathbf{r}_d
\end{aligned} \tag{11.89}$$

The second term is:

$$\begin{aligned}
\int_A \delta \hat{\boldsymbol{\epsilon}}^T \boldsymbol{\sigma} dA &= \\
&= \delta \boldsymbol{\beta}^T \sum_{i=1}^4 \mathbf{P} (\boldsymbol{\xi}_i)^T \boldsymbol{\sigma} (\boldsymbol{\xi}_i) \alpha (\boldsymbol{\xi}_i) w_i \\
&= \delta \boldsymbol{\beta}^T \mathbf{r}_\beta
\end{aligned} \tag{11.90}$$

The resulting governing equations are then:

$$(\mathbf{K}_m + \mathbf{K}_g) \Delta \mathbf{q} + \mathbf{K}_{\beta q}^T \Delta \boldsymbol{\beta} = -\mathbf{r}_d \tag{11.91a}$$

$$\mathbf{K}_{\beta q} \Delta \mathbf{q} + \mathbf{K}_{\beta \beta} \Delta \boldsymbol{\beta} = -\mathbf{r}_\beta \tag{11.91b}$$

Chapter 12

Aerodynamic Elements

Aerodynamic elements apply aerodynamic forces to structural nodes.

This section is not intended to give details about the aerodynamic models adopted but mainly discuss the computation of the contributions to the Jacobian matrix of the aerodynamic elements.

Files. 2D aerodynamic elements are implemented in files

`mbdyn/aero/aeroelem.h`

`mbdyn/aero/aeroelem.cc`

2D aerodynamic models are implemented in files

`mbdyn/aero/aerodata.h`

`mbdyn/aero/aerodata.cc`

`mbdyn/aero/aerodata_impl.h`

`mbdyn/aero/aerodata_impl.cc`

12.1 Linearization of 2D Aerodynamic Forces and Moments

MBDyn's built-in 2D aerodynamics computes aerodynamic forces $\tilde{\mathbf{f}}_{a/\xi}$ and moments $\tilde{\mathbf{c}}_{a/\xi}$ per unit span, in a relative frame at station ξ , based on the instantaneous value of linear and angular velocity boundary conditions, respectively $\tilde{\mathbf{v}}$ and $\tilde{\boldsymbol{\omega}}$, expressed in the same relative frame, namely

$$\tilde{\mathbf{f}}_{a/\xi} = \tilde{\mathbf{f}}_{a/\xi}(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}, \xi) \quad (12.1a)$$

$$\tilde{\mathbf{c}}_{a/\xi} = \tilde{\mathbf{c}}_{a/\xi}(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}, \xi). \quad (12.1b)$$

The boundary conditions are computed by projecting the *effective* linear and angular velocity at station ξ , respectively $\tilde{\mathbf{v}}(\xi)$ and $\tilde{\boldsymbol{\omega}}(\xi)$, in the reference frame of the aerodynamic forces, namely

$$\tilde{\mathbf{v}}(\xi) = \mathbf{R}^T(\xi) \mathbf{v}(\xi) \quad (12.2a)$$

$$\tilde{\boldsymbol{\omega}}(\xi) = \mathbf{R}^T(\xi) \boldsymbol{\omega}(\xi), \quad (12.2b)$$

where $\mathbf{R}(\xi)$ is the matrix that expresses the local orientation of the aerodynamic section at station ξ .

In detail, the effective linear velocity at an arbitrary station is the combination of the absolute velocity resulting from the kinematics of the model, of an airstream velocity that may depend on the absolute location of a reference point and on time, and of a contribution resulting from an inflow model, namely

$$\mathbf{v}(\xi) = \mathbf{v}_{\text{kin}}(\xi) + \mathbf{v}_\infty(\mathbf{x}(\xi), t) + \mathbf{v}_{\text{inflow}}(\mathbf{x}(\xi)). \quad (12.3)$$

It is assumed that the last two contributions do not depend on the state of the problem, or only depend on it in a loose manner, and thus do not directly participate in the linearization of the aerodynamic forces and moments.

Their linearization yields

$$\delta\tilde{\mathbf{v}} = \mathbf{R}^T(\xi)(\delta\mathbf{v}_{\text{kin}}(\xi) + \mathbf{v}(\xi) \times \boldsymbol{\theta}_\delta(\xi)) \quad (12.4a)$$

$$\delta\tilde{\boldsymbol{\omega}} = \mathbf{R}^T(\xi)(\delta\boldsymbol{\omega}(\xi) + \boldsymbol{\omega}(\xi) \times \boldsymbol{\theta}_\delta(\xi)), \quad (12.4b)$$

where

$$\boldsymbol{\theta}_\delta(\xi) = \text{ax}(\delta\mathbf{R}(\xi)\mathbf{R}^T(\xi)). \quad (12.5)$$

By means of numerical integration, the force and moment per unit span are integrated into discrete contributions to the force and the moment applied to the appropriate node equilibrium. This is usually done by multiplying each force and moment per unit span contribution by an appropriate reference length coefficient. As a consequence, in the following force and moment contributions will be considered, namely

$$\tilde{\mathbf{f}}_a = \tilde{\mathbf{f}}_a(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}, \xi) \quad (12.6a)$$

$$\tilde{\mathbf{c}}_a = \tilde{\mathbf{c}}_a(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}, \xi). \quad (12.6b)$$

As already mentioned, each weighted sectional force and moment contribution is applied to the appropriate node after projection in the global reference frame. The contribution to the force and moment equilibrium of the n -th node is

$$\Delta\mathbf{f}_n = \mathbf{R}(\xi)\tilde{\mathbf{f}}_a \quad (12.7a)$$

$$\Delta\mathbf{c}_n = \mathbf{R}(\xi)\tilde{\mathbf{c}}_a + (\mathbf{x}(\xi) - \mathbf{x}_n) \times \Delta\mathbf{f}_n. \quad (12.7b)$$

Their linearization yields

$$\delta\Delta\mathbf{f}_n = -\Delta\mathbf{f}_n \times \boldsymbol{\theta}_\delta(\xi) + \mathbf{R}(\xi)\delta\tilde{\mathbf{f}}_a \quad (12.8a)$$

$$\begin{aligned} \delta\Delta\mathbf{c}_n = & -\Delta\mathbf{c}_n \times \boldsymbol{\theta}_\delta(\xi) + \mathbf{R}(\xi)\delta\tilde{\mathbf{c}}_a - \Delta\mathbf{f}_n \times (\delta\mathbf{x}(\xi) - \delta\mathbf{x}_n) \\ & + (\mathbf{x}(\xi) - \mathbf{x}_n) \times (-\Delta\mathbf{f}_n \times \boldsymbol{\theta}_\delta(\xi) + \mathbf{R}(\xi)\delta\tilde{\mathbf{f}}_a), \end{aligned} \quad (12.8b)$$

which can be summarized as

$$\begin{aligned} \left\{ \begin{array}{l} \delta\Delta\mathbf{f}_n \\ \delta\Delta\mathbf{c}_n \end{array} \right\} = & \left[\begin{array}{c} -\Delta\mathbf{f}_n \times \\ -\Delta\mathbf{c}_n \times -(\mathbf{x}(\xi) - \mathbf{x}_n) \times \Delta\mathbf{f}_n \times \end{array} \right] \boldsymbol{\theta}_\delta(\xi) \\ & + \left[\begin{array}{c} \mathbf{0} \\ -\Delta\mathbf{f}_n \times \end{array} \right] (\delta\mathbf{x}(\xi) - \delta\mathbf{x}_n) \\ & + \left[\begin{array}{cc} \mathbf{I} & \mathbf{0} \\ (\mathbf{x}(\xi) - \mathbf{x}_n) \times & \mathbf{I} \end{array} \right] \left\{ \begin{array}{l} \mathbf{R}(\xi)\delta\tilde{\mathbf{f}}_a \\ \mathbf{R}(\xi)\delta\tilde{\mathbf{c}}_a \end{array} \right\}. \end{aligned} \quad (12.9)$$

It is assumed that the Jacobian matrix of the sectional force and moment with respect to the linear and angular velocity is either available or can be computed by numerical differentiation. The resulting force and moment perturbation is

$$\left\{ \begin{array}{l} \delta\tilde{\mathbf{f}}_a \\ \delta\tilde{\mathbf{c}}_a \end{array} \right\} = \left[\begin{array}{cc} \tilde{\mathbf{f}}_{a/\tilde{\mathbf{v}}} & \tilde{\mathbf{f}}_{a/\tilde{\boldsymbol{\omega}}} \\ \tilde{\mathbf{c}}_{a/\tilde{\mathbf{v}}} & \tilde{\mathbf{c}}_{a/\tilde{\boldsymbol{\omega}}} \end{array} \right] \left\{ \begin{array}{l} \delta\tilde{\mathbf{v}} \\ \delta\tilde{\boldsymbol{\omega}} \end{array} \right\}. \quad (12.10)$$

The computation of the matrix of Eq. (12.10) is delegated to the `AeroData` class.

Each type of element determines how the sectional force and moment contributions are applied to the nodes, and how the sectional boundary conditions at each section are computed from the kinematics of the nodes.

12.2 Numerical Linearization of Sectional Forces

Consider an arbitrary submatrix of the Jacobian matrix of Eq. (12.10), $\mathbf{J} = \mathbf{p}_{/\mathbf{q}}$. Its generic element, the c -th component of \mathbf{p} derived by the r -th component of \mathbf{q} , is

$$J_{rc} = \frac{\partial \mathbf{p}_r}{\partial q_c}. \quad (12.11)$$

A forward difference approach is used, namely

$$J_{rc} \cong \frac{\mathbf{p}_r(\mathbf{q} + \Delta q \mathbf{e}_c) - \mathbf{p}_r(\mathbf{q})}{\Delta q}, \quad (12.12)$$

where \mathbf{e}_c is the unit vector along the c -th component, and Δq is a suitably chosen perturbation. Alternatively, a centered difference approach can be used, namely

$$J_{rc} \cong \frac{\mathbf{p}_r(\mathbf{q} + \Delta q \mathbf{e}_c) - \mathbf{p}_r(\mathbf{q} - \Delta q \mathbf{e}_c)}{2\Delta q}. \quad (12.13)$$

The perturbation is

$$\Delta q = \varepsilon \|\mathbf{q}\| + \nu. \quad (12.14)$$

Since the boundary condition \mathbf{q} is perturbed in order to determine an equivalent perturbation of angle of attack, a resolution of few tenth of degree is deemed sufficient. As a consequence, $\varepsilon > 0$ must be a “small” number that, in case \mathbf{e}_c is orthogonal to \mathbf{q} , yields an angle of the order of a tenth of a degree. The default value is $\varepsilon = 10^{-3}$. However, in order to avoid divisions by too small numbers, the perturbation is corrected by another “small” parameter, $\nu > 0$. The default value is $\nu = 10^{-9}$.

12.3 Aerodynamic Forces with Internal States

Consider the case of a model of the aerodynamic forces that requires the use of internal states \mathbf{q} , namely

$$\tilde{\mathbf{f}}_a = \tilde{\mathbf{f}}_a(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}, \mathbf{q}) \quad (12.15a)$$

$$\tilde{\mathbf{c}}_a = \tilde{\mathbf{c}}_a(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}, \mathbf{q}) \quad (12.15b)$$

$$\mathbf{0} = \mathbf{g}(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}, \mathbf{q}, \dot{\mathbf{q}}). \quad (12.15c)$$

Usually, the dynamic model of the aerodynamics is differential, and thus its perturbation yields a linearized state-space system,

$$\delta \mathbf{g} = \mathbf{g}_{/\tilde{\mathbf{v}}} \delta \tilde{\mathbf{v}} + \mathbf{g}_{/\tilde{\boldsymbol{\omega}}} \delta \tilde{\boldsymbol{\omega}} + \mathbf{g}_{/\mathbf{q}} \delta \mathbf{q} + \mathbf{g}_{/\dot{\mathbf{q}}} \delta \dot{\mathbf{q}} \quad (12.16a)$$

$$\delta \tilde{\mathbf{f}}_a = \tilde{\mathbf{f}}_{a/\tilde{\mathbf{v}}} \delta \tilde{\mathbf{v}} + \tilde{\mathbf{f}}_{a/\tilde{\boldsymbol{\omega}}} \delta \tilde{\boldsymbol{\omega}} + \tilde{\mathbf{f}}_{a/\mathbf{q}} \delta \mathbf{q} \quad (12.16b)$$

$$\delta \tilde{\mathbf{c}}_a = \tilde{\mathbf{c}}_{a/\tilde{\mathbf{v}}} \delta \tilde{\mathbf{v}} + \tilde{\mathbf{c}}_{a/\tilde{\boldsymbol{\omega}}} \delta \tilde{\boldsymbol{\omega}} + \tilde{\mathbf{c}}_{a/\mathbf{q}} \delta \mathbf{q}, \quad (12.16c)$$

where, with respect to $\delta \mathbf{g}$, the terms $\delta \tilde{\mathbf{v}}$ and $\delta \tilde{\boldsymbol{\omega}}$ play the role of the input, while $\delta \mathbf{q}$ plays the role of the state; $\delta \tilde{\mathbf{f}}_a$, $\delta \tilde{\mathbf{c}}_a$ play the role of the output.

In those cases, the underlying aerodynamic model has to deal with \mathbf{g} , Eq. (12.15c), and its perturbation $\delta \mathbf{g}$, Eq. (12.15c).

However, the aerodynamic elements have to:

1. provide the underlying aerodynamic model the Jacobian submatrices required to compute $\delta \tilde{\mathbf{v}}$ and $\delta \tilde{\boldsymbol{\omega}}$ from the perturbations of the nodal position, orientation, and linear and angular velocity that are needed to deal with $\delta \mathbf{g}$, Eq. (12.15c);
2. account for the contribution of the Jacobian matrices $\delta \tilde{\mathbf{f}}_{a/\mathbf{q}}$ and $\delta \tilde{\mathbf{c}}_{a/\mathbf{q}}$ respectively required by Eqs. (12.16b) and (12.16c).

12.4 Aerodynamic body

The aerodynamic body element applies aerodynamic forces to the structural node it is connected to, based on the relative velocity between an aerodynamic surface attached to the node and the airstream.

The boundary conditions are related to the rigid body motion of the node, so

$$\mathbf{R}(\xi) = \mathbf{R}_n \mathbf{R}_a \mathbf{R}_t(\xi) \quad (12.17a)$$

$$\mathbf{b}(\xi) = \mathbf{R}_n (\tilde{\mathbf{b}}_0 + \mathbf{R}_a (b(\xi) \mathbf{e}_1 + \xi \mathbf{e}_3)) \quad (12.17b)$$

$$\boldsymbol{\omega}(\xi) = \boldsymbol{\omega}_n \quad (12.17c)$$

$$\mathbf{v}_{\text{kin}}(\xi) = \mathbf{v}_n + \boldsymbol{\omega}_n \times \mathbf{b}(\xi), \quad (12.17d)$$

where \mathbf{R}_n is the orientation of the node, \mathbf{R}_a is the relative orientation of the aerodynamics with respect to the node, \mathbf{R}_t is the pretwist matrix, \mathbf{v}_n is the absolute velocity of the node, $\boldsymbol{\omega}_n$ is the absolute angular velocity of the node, $\tilde{\mathbf{b}}_0$ is an offset between the node and the reference location of the aerodynamic body, and $b(\xi)$ is the chordwise location of the point where the boundary conditions are evaluated.

Their linearization is straightforward:

$$\boldsymbol{\theta}_\delta(\xi) = \boldsymbol{\theta}_{n\delta} \quad (12.18a)$$

$$\delta \mathbf{b}(\xi) = -\mathbf{b}(\xi) \times \boldsymbol{\theta}_{n\delta} \quad (12.18b)$$

$$\delta \boldsymbol{\omega}(\xi) = \delta \boldsymbol{\omega}_n \quad (12.18c)$$

$$\delta \mathbf{v}_{\text{kin}}(\xi) = \delta \mathbf{v}_n - \mathbf{b}(\xi) \times \delta \boldsymbol{\omega}_n - \boldsymbol{\omega}_n \times \mathbf{b}(\xi) \times \boldsymbol{\theta}_{n\delta}. \quad (12.18d)$$

Eq. (12.7b) can be rewritten as

$$\Delta \mathbf{c}_n = \mathbf{R}(\xi) (\tilde{\mathbf{c}}_a + \tilde{\mathbf{o}}(\xi) \times \tilde{\mathbf{f}}_a), \quad (12.19)$$

where $\tilde{\mathbf{o}}(\xi) = \mathbf{R}^T(\xi) (\mathbf{x}(\xi) - \mathbf{x}_n)$, the offset between the point where the force is applied and the node, in the reference frame of the node, does not depend on the kinematics of the system, since the body is rigid. Its linearization yields

$$\delta \Delta \mathbf{c}_n = -\Delta \mathbf{c}_n \times \boldsymbol{\theta}_\delta(\xi) + \mathbf{R}(\xi) (\delta \tilde{\mathbf{c}}_a + \tilde{\mathbf{o}}(\xi) \times \delta \tilde{\mathbf{f}}_a), \quad (12.20)$$

since $\delta \tilde{\mathbf{o}}(\xi) \equiv 0$. So the linearized force and moment is

$$\begin{Bmatrix} \delta \Delta \mathbf{f}_n \\ \delta \Delta \mathbf{c}_n \end{Bmatrix} = - \begin{bmatrix} \Delta \mathbf{f}_n \times \\ \Delta \mathbf{c}_n \times \end{bmatrix} \boldsymbol{\theta}_{n\delta} + \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{o}(\xi) \times & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{R}(\xi) \delta \tilde{\mathbf{f}}_a \\ \mathbf{R}(\xi) \delta \tilde{\mathbf{c}}_a \end{Bmatrix}. \quad (12.21)$$

with $\mathbf{o}(\xi) = \mathbf{R}(\xi) \tilde{\mathbf{o}}(\xi)$.

The linearization of the boundary conditions yields

$$\begin{Bmatrix} \delta \tilde{\mathbf{v}}(\xi) \\ \delta \tilde{\boldsymbol{\omega}}(\xi) \end{Bmatrix} = \begin{bmatrix} \mathbf{R}^T(\xi) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^T(\xi) \end{bmatrix} \left(\begin{bmatrix} \mathbf{v}(\xi) \times & -\boldsymbol{\omega}_n \times \mathbf{b}(\xi) \times \\ \boldsymbol{\omega}_n \times & \end{bmatrix} \boldsymbol{\theta}_{n\delta} \right. \\ \left. + \begin{bmatrix} \mathbf{I} & -\mathbf{b}(\xi) \times \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \delta \mathbf{v}_n \\ \delta \boldsymbol{\omega}_n \end{Bmatrix} \right) \quad (12.22a)$$

$$\begin{aligned} &\stackrel{\text{uu}}{=} \begin{bmatrix} \mathbf{R}^T(\xi) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^T(\xi) \end{bmatrix} \left(\begin{bmatrix} (\mathbf{v}(\xi) + \mathbf{b}(\xi) \times \boldsymbol{\omega}_n) \times & \mathbf{0} \\ \mathbf{0} & \end{bmatrix} \delta \mathbf{g}_n \right. \\ &\quad \left. + \begin{bmatrix} \mathbf{I} & -\mathbf{b}(\xi) \times \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \delta \dot{\mathbf{x}}_n \\ \delta \dot{\mathbf{g}}_n \end{Bmatrix} \right). \end{aligned} \quad (12.22b)$$

After defining

$$\mathbf{f}_{a/\tilde{\mathbf{v}}} = \mathbf{R}(\xi) \tilde{\mathbf{f}}_{a/\tilde{\mathbf{v}}} \mathbf{R}^T(\xi) \quad (12.23a)$$

$$\mathbf{f}_{a/\tilde{\omega}} = \mathbf{R}(\xi) \tilde{\mathbf{f}}_{a/\tilde{\omega}} \mathbf{R}^T(\xi) \quad (12.23b)$$

$$\mathbf{c}_{a/\tilde{\mathbf{v}}} = \mathbf{R}(\xi) \tilde{\mathbf{c}}_{a/\tilde{\mathbf{v}}} \mathbf{R}^T(\xi) \quad (12.23c)$$

$$\mathbf{c}_{a/\tilde{\omega}} = \mathbf{R}(\xi) \tilde{\mathbf{c}}_{a/\tilde{\omega}} \mathbf{R}^T(\xi), \quad (12.23d)$$

the linearization becomes

$$\begin{aligned} & \left[\mathbf{o}(\xi) \times \mathbf{f}_{a/\tilde{\mathbf{v}}} + \mathbf{c}_{a/\tilde{\mathbf{v}}} \right] \delta \dot{\mathbf{x}}_n \\ & + \left[\mathbf{o}(\xi) \times \mathbf{f}_{a/\tilde{\omega}} + \mathbf{c}_{a/\tilde{\omega}} - (\mathbf{o}(\xi) \times \mathbf{f}_{a/\tilde{\mathbf{v}}} + \mathbf{c}_{a/\tilde{\mathbf{v}}}) \mathbf{b}(\xi) \times \right] \delta \dot{\mathbf{g}}_n \\ & + \left[(\mathbf{o}(\xi) \times \mathbf{f}_{a/\tilde{\mathbf{v}}} + \mathbf{c}_{a/\tilde{\mathbf{v}}}) (\mathbf{v}(\xi) + \mathbf{b}(\xi) \times \boldsymbol{\omega}_n) \times \right] \delta \mathbf{g}_n \\ & - \left[\begin{array}{c} \Delta \mathbf{f}_n \times \\ \Delta \mathbf{c}_n \times \end{array} \right] \delta \mathbf{g}_n \stackrel{\text{uu}}{=} \left\{ \begin{array}{c} \delta \Delta \mathbf{f}_n \\ \delta \Delta \mathbf{c}_n \end{array} \right\}. \end{aligned} \quad (12.24)$$

12.5 Aerodynamic Beam (3 Nodes)

The `aerodynamic beam3` element applies aerodynamic forces to the nodes of a three node structural beam element.

The aerodynamic forces/momenta acting on each node are computed based on the relative velocity of a set of locations along the beam and the airstream. The kinematic quantities of the beam are computed based on an interpolation of the kinematics of the three nodes.

Kinematics Interpolation. *Note: this part is common to all elements that use the three-node beam discretization and interpolation model.* The generic field variable $\mathbf{p}(x)$ is interpolated using parabolic functions related to the value of the field variable at three locations that in general can be offset from the nodes,

$$\mathbf{p}(\xi) = \sum_{i=1,2,3} N_i(\xi) \mathbf{p}_i, \quad (12.25)$$

with

$$N_1 = \frac{1}{2} \xi (\xi - 1) \quad (12.26a)$$

$$N_2 = 1 - \xi^2 \quad (12.26b)$$

$$N_3 = \frac{1}{2} \xi (\xi + 1). \quad (12.26c)$$

Orientation at an arbitrary location ξ :

$$\mathbf{R}(\xi) = \mathbf{R}_{2_a} \mathbf{R}(\boldsymbol{\theta}(\xi)) \quad (12.27)$$

$$\boldsymbol{\theta}(\xi) = \sum_{i=1,3} N_i(\xi) \boldsymbol{\theta}_{2 \leftarrow i} = N_1(\xi) \boldsymbol{\theta}_{2 \leftarrow 1} + N_3(\xi) \boldsymbol{\theta}_{2 \leftarrow 3} \quad (12.28)$$

$$\boldsymbol{\theta}_{2 \leftarrow i} = \text{ax}(\exp^{-1}(\mathbf{R}_{2_a}^T \mathbf{R}_{i_a})). \quad (12.29)$$

Orientation is dealt with specially, given its special nature. The orientation of the mid node is used as a reference, and the orientation parameters that express the relative orientation between each of the end nodes and the mid node are interpolated. The interpolated orientation parameters are used to compute the interpolated relative orientation matrix, which is then pre-multiplied by the orientation matrix of the mid node. Summation in this case occurs on $i = 1, 3$ only because by definition $\boldsymbol{\theta}_{2 \leftarrow 2} = \text{ax}(\exp^{-1}((\mathbf{I})) \equiv \mathbf{0}$.

Since the Euler-Rodrigues orientation parameters are used (the so-called ‘rotation vector’), the magnitude of the relative orientation between each end node and the mid node must be limited (formally, to π , but it should be less for accuracy).

Position at an arbitrary location ξ :

$$\mathbf{x}(\xi) = \sum_{i=1,2,3} N_i(\xi) (\mathbf{x}_i + \mathbf{o}_i), \quad (12.30)$$

with $\mathbf{o}_i = \mathbf{R}_i \tilde{\mathbf{o}}_i$.

Angular velocity at an arbitrary location ξ :

$$\boldsymbol{\omega}(\xi) = \sum_{i=1,2,3} N_i(\xi) \boldsymbol{\omega}_i. \quad (12.31)$$

Velocity at an arbitrary location ξ :

$$\mathbf{v}_{\text{kin}}(\xi) = \sum_{i=1,2,3} N_i(\xi) (\mathbf{v}_i + \boldsymbol{\omega}_i \times \mathbf{o}_i). \quad (12.32)$$

Perturbation of Interpolated Kinematics. Orientation perturbation at an arbitrary location ξ :

$$\delta\boldsymbol{\theta}(\xi) = \sum_{i=1,3} N_i(\xi) \delta\boldsymbol{\theta}_{2 \leftarrow i} \quad (12.33)$$

$$\delta\boldsymbol{\theta}_{2 \leftarrow i} = \boldsymbol{\Gamma}^{-1}(\boldsymbol{\theta}_{2 \leftarrow i}) \boldsymbol{\theta}_{(2 \leftarrow i)\delta} \quad (12.34)$$

$$\boldsymbol{\theta}_{(2 \leftarrow i)\delta} = \mathbf{R}_{2_a}^T (\boldsymbol{\theta}_{i\delta} - \boldsymbol{\theta}_{2\delta}) \quad (12.35)$$

$$\begin{aligned} \boldsymbol{\theta}_\delta(\xi) &= \boldsymbol{\theta}_{2\delta} + \sum_{i=1,3} \mathbf{R}_{2_a} \boldsymbol{\Gamma}(\boldsymbol{\theta}(\xi)) N_i(\xi) \boldsymbol{\Gamma}^{-1}(\boldsymbol{\theta}_{2 \leftarrow i}) \mathbf{R}_{2_a}^T (\boldsymbol{\theta}_{i\delta} - \boldsymbol{\theta}_{2\delta}) \\ &= \left(\mathbf{I} - \mathbf{R}_{2_a} \boldsymbol{\Gamma}(\boldsymbol{\theta}(\xi)) \sum_{i=1,3} N_i(\xi) \boldsymbol{\Gamma}^{-1}(\boldsymbol{\theta}_{2 \leftarrow i}) \mathbf{R}_{2_a}^T \right) \boldsymbol{\theta}_{2\delta} \\ &\quad + \mathbf{R}_{2_a} \boldsymbol{\Gamma}(\boldsymbol{\theta}(\xi)) N_1 \boldsymbol{\Gamma}^{-1}(\boldsymbol{\theta}_{2 \leftarrow 1}) \mathbf{R}_{2_a}^T \boldsymbol{\theta}_{1\delta} \\ &\quad + \mathbf{R}_{2_a} \boldsymbol{\Gamma}(\boldsymbol{\theta}(\xi)) N_3 \boldsymbol{\Gamma}^{-1}(\boldsymbol{\theta}_{2 \leftarrow 3}) \mathbf{R}_{2_a}^T \boldsymbol{\theta}_{3\delta} \\ &= \sum_{i=1,2,3} \boldsymbol{\Theta}_i(\xi) \boldsymbol{\theta}_{i\delta}, \end{aligned} \quad (12.36)$$

with

$$\boldsymbol{\Theta}_i(\xi) = \mathbf{R}_{2_a} \boldsymbol{\Gamma}(\boldsymbol{\theta}(\xi)) N_i \boldsymbol{\Gamma}^{-1}(\boldsymbol{\theta}_{2 \leftarrow i}) \mathbf{R}_{2_a}^T \quad i = 1 \text{ and } 3 \quad (12.37a)$$

$$\boldsymbol{\Theta}_2(\xi) = \mathbf{I} - \boldsymbol{\Theta}_1(\xi) - \boldsymbol{\Theta}_3(\xi) \quad (12.37b)$$

playing the role of shape functions.

In fact, note that, when $N_1 = 1$ and $N_3 = 0$, then $\boldsymbol{\Theta}_1 = \mathbf{I}$, $\boldsymbol{\Theta}_2 = \boldsymbol{\Theta}_3 = \mathbf{0}$ and $\boldsymbol{\theta}_\delta(\xi) = \boldsymbol{\theta}_{1\delta}$, while, when $N_1 = 0$ and $N_3 = 1$, then $\boldsymbol{\Theta}_1 = \boldsymbol{\Theta}_2 = \mathbf{0}$, $\boldsymbol{\Theta}_3 = \mathbf{I}$ and $\boldsymbol{\theta}_\delta(\xi) = \boldsymbol{\theta}_{3\delta}$. Finally, when $N_1 = N_3 = 0$, then $\boldsymbol{\Theta}_1 = \boldsymbol{\Theta}_3 = \mathbf{0}$, $\boldsymbol{\Theta}_2 = \mathbf{I}$ and $\boldsymbol{\theta}_\delta(\xi) = \boldsymbol{\theta}_{2\delta}$. Moreover, $\sum_{i=1,2,3} \boldsymbol{\Theta}_i(\xi) = \mathbf{I} \forall \xi$.

Position perturbation at an arbitrary location ξ :

$$\delta\mathbf{x}(\xi) = \sum_{i=1,2,3} N_i(\xi) (\delta\mathbf{x}_i + \boldsymbol{\theta}_{i\delta} \times \mathbf{o}_i). \quad (12.38)$$

Angular velocity perturbation at an arbitrary location ξ :

$$\begin{aligned} \delta\boldsymbol{\omega}(\xi) &= \sum_{i=1,2,3} N_i \delta\boldsymbol{\omega}_i \\ &\stackrel{\text{uu}}{=} \sum_{i=1,2,3} N_i (\delta\dot{\mathbf{g}}_i - \boldsymbol{\omega}_i \times \delta\mathbf{g}_i). \end{aligned} \quad (12.39)$$

Velocity perturbation at an arbitrary location ξ :

$$\begin{aligned} \delta\mathbf{v}_{\text{kin}}(\xi) &= \sum_{i=1,2,3} N_i(\xi) (\delta\mathbf{v}_i + \delta\boldsymbol{\omega}_i \times \mathbf{o}_i + \boldsymbol{\omega}_i \times \delta\mathbf{o}_i) \\ &= \sum_{i=1,2,3} N_i(\xi) (\delta\mathbf{v}_i - \mathbf{o}_i \times \delta\boldsymbol{\omega}_i - \boldsymbol{\omega}_i \times \mathbf{o}_i \times \boldsymbol{\theta}_{i\delta}) \\ &\stackrel{\text{uu}}{=} \sum_{i=1,2,3} N_i(\xi) (\delta\dot{\mathbf{x}}_i - \mathbf{o}_i \times \delta\dot{\mathbf{g}}_i - (\boldsymbol{\omega}_i \times \mathbf{o}_i) \times \delta\mathbf{g}_i). \end{aligned} \quad (12.40)$$

Boundary Conditions Perturbation. Angular velocity perturbation:

$$\begin{aligned} \delta\tilde{\boldsymbol{\omega}}(\xi) &= \mathbf{R}^T(\xi) \sum_{i=1,2,3} (N_i(\xi) \delta\boldsymbol{\omega}_i + \boldsymbol{\omega}(\xi) \times \boldsymbol{\Theta}_i(\xi) \boldsymbol{\theta}_{i\delta}) \\ &\stackrel{\text{uu}}{=} \mathbf{R}^T(\xi) (N_i(\xi) \delta\dot{\mathbf{g}}_i + (\boldsymbol{\omega}(\xi) \times \boldsymbol{\Theta}_i(\xi) - N_i(\xi) \boldsymbol{\omega}_i \times) \delta\mathbf{g}_i). \end{aligned} \quad (12.41)$$

Velocity perturbation:

$$\begin{aligned} \delta\tilde{\mathbf{v}}(\xi) &= \mathbf{R}^T(\xi) (N_i(\xi) \delta\mathbf{v}_i - N_i(\xi) \mathbf{o}_i \times \delta\boldsymbol{\omega}_i \\ &\quad + (\boldsymbol{v}(\xi) \times \boldsymbol{\Theta}_i(\xi) - N_i(\xi) \boldsymbol{\omega}_i \times \mathbf{o}_i \times) \boldsymbol{\theta}_{i\delta}) \\ &\stackrel{\text{uu}}{=} \mathbf{R}^T(\xi) (N_i(\xi) \delta\dot{\mathbf{x}}_i - N_i(\xi) \mathbf{o}_i \times \delta\dot{\mathbf{g}}_i \\ &\quad + (\boldsymbol{v}(\xi) \times \boldsymbol{\Theta}_i(\xi) - N_i(\xi) (\boldsymbol{\omega}_i \times \mathbf{o}_i) \times) \delta\mathbf{g}_i). \end{aligned} \quad (12.42)$$

They can be summarized as

$$\begin{aligned} \left\{ \begin{array}{c} \delta\tilde{\mathbf{v}}(\xi) \\ \delta\tilde{\boldsymbol{\omega}}(\xi) \end{array} \right\} &= \left[\begin{array}{cc} \mathbf{R}^T(\xi) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^T(\xi) \end{array} \right] \sum_{i=1,2,3} \left(N_i(\xi) \begin{bmatrix} \mathbf{I} & -\mathbf{o}_i \times \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \right) \left\{ \begin{array}{c} \delta\mathbf{v}_i \\ \delta\boldsymbol{\omega}_i \end{array} \right\} \\ &\quad + \left[\begin{array}{c} \boldsymbol{v}(\xi) \times \boldsymbol{\Theta}_i(\xi) - N_i(\xi) \boldsymbol{\omega}_i \times \mathbf{o}_i \times \\ \boldsymbol{\omega}(\xi) \times \boldsymbol{\Theta}_i(\xi) \end{array} \right] \boldsymbol{\theta}_{i\delta} \\ &\stackrel{\text{uu}}{=} \left[\begin{array}{cc} \mathbf{R}^T(\xi) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^T(\xi) \end{array} \right] \sum_{i=1,2,3} \left(N_i(\xi) \begin{bmatrix} \mathbf{I} & -\mathbf{o}_i \times \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \right) \left\{ \begin{array}{c} \delta\dot{\mathbf{x}}_i \\ \delta\dot{\mathbf{g}}_i \end{array} \right\} \\ &\quad + \left[\begin{array}{c} \boldsymbol{v}(\xi) \times \boldsymbol{\Theta}_i(\xi) - N_i(\xi) (\boldsymbol{\omega}_i \times \mathbf{o}_i) \times \\ \boldsymbol{\omega}(\xi) \times \boldsymbol{\Theta}_i(\xi) - N_i(\xi) \boldsymbol{\omega}_i \times \end{array} \right] \delta\mathbf{g}_i. \end{aligned} \quad (12.43)$$

Contribution to Jacobian Matrix. After defining

$$\mathbf{f}_{a/\tilde{\mathbf{v}}} = \mathbf{R}(\xi) \tilde{\mathbf{f}}_{a/\tilde{\mathbf{v}}} \mathbf{R}^T(\xi) \quad (12.44a)$$

$$\mathbf{f}_{a/\tilde{\omega}} = \mathbf{R}(\xi) \tilde{\mathbf{f}}_{a/\tilde{\omega}} \mathbf{R}^T(\xi) \quad (12.44b)$$

$$\mathbf{c}_{a/\tilde{\mathbf{v}}} = \mathbf{R}(\xi) \tilde{\mathbf{c}}_{a/\tilde{\mathbf{v}}} \mathbf{R}^T(\xi) \quad (12.44c)$$

$$\mathbf{c}_{a/\tilde{\omega}} = \mathbf{R}(\xi) \tilde{\mathbf{c}}_{a/\tilde{\omega}} \mathbf{R}^T(\xi) \quad (12.44d)$$

$$\mathbf{B}_{\tilde{\mathbf{v}}} = \mathbf{v}(\xi) \times \Theta_i(\xi) - N_i(\xi) (\boldsymbol{\omega}_i \times \mathbf{o}_i) \times \quad (12.44e)$$

$$\mathbf{B}_{\tilde{\omega}} = \boldsymbol{\omega}(\xi) \times \Theta_i(\xi) - N_i(\xi) \boldsymbol{\omega}_i \times \quad (12.44f)$$

$$\mathbf{d}_n(\xi) = \mathbf{x}(\xi) - \mathbf{x}_n, \quad (12.44g)$$

the contribution of the i -th node's motion to the equilibrium of the n -th node ($i = 1, 2, 3$, $n = 1, 2, 3$) is

$$\begin{aligned} & N_i(\xi) \left[\mathbf{d}_n(\xi) \times \frac{\mathbf{f}_{a/\tilde{\mathbf{v}}}}{\mathbf{f}_{a/\tilde{\mathbf{v}}} + \mathbf{c}_{a/\tilde{\mathbf{v}}}} \right] \delta \dot{\mathbf{x}}_i \\ & + N_i(\xi) \left[\mathbf{d}_n(\xi) \times \frac{\mathbf{f}_{a/\tilde{\omega}} - \mathbf{f}_{a/\tilde{\mathbf{v}}} \mathbf{o}_i \times}{\mathbf{d}_n(\xi) \times \mathbf{f}_{a/\tilde{\omega}} + \mathbf{c}_{a/\tilde{\omega}} - (\mathbf{d}_n(\xi) \times \mathbf{f}_{a/\tilde{\mathbf{v}}} + \mathbf{c}_{a/\tilde{\mathbf{v}}}) \mathbf{o}_i \times} \right] \delta \dot{\mathbf{g}}_i \\ & + \left[\mathbf{d}_n(\xi) \times \left(\frac{\mathbf{f}_{a/\tilde{\mathbf{v}}} \mathbf{B}_{\tilde{\mathbf{v}}} + \mathbf{f}_{a/\tilde{\omega}} \mathbf{B}_{\tilde{\omega}}}{\mathbf{d}_n(\xi) \times (\mathbf{f}_{a/\tilde{\mathbf{v}}} \mathbf{B}_{\tilde{\mathbf{v}}} + \mathbf{f}_{a/\tilde{\omega}} \mathbf{B}_{\tilde{\omega}}) + \mathbf{c}_{a/\tilde{\mathbf{v}}} \mathbf{B}_{\tilde{\mathbf{v}}} + \mathbf{c}_{a/\tilde{\omega}} \mathbf{B}_{\tilde{\omega}}} \right) \delta \mathbf{g}_i \right. \\ & \quad \left. + \left[\begin{array}{c} \mathbf{0} \\ -(N_i - \delta_{ni}) \Delta \mathbf{f}_n \times \end{array} \right] \delta \mathbf{x}_i \right. \\ & + \left[\begin{array}{c} -\Delta \mathbf{f}_n \times \Theta_i(\xi) \\ -(\mathbf{d}_n(\xi) \times \Delta \mathbf{f}_n \times + \Delta \mathbf{c}_n \times) \Theta_i(\xi) + \Delta \mathbf{f}_n \times N_i \mathbf{o}_i \times \end{array} \right] \delta \mathbf{g}_i \\ & \stackrel{\text{uu}}{=} \left\{ \begin{array}{c} \delta \Delta \mathbf{f}_n \\ \delta \Delta \mathbf{c}_n \end{array} \right\}, \end{aligned} \quad (12.45)$$

where δ_{ni} is Dirac's delta, which is 1 when $i = n$, and 0 otherwise.

12.6 Aerodynamic Beam (2 Nodes)

The `aerodynamic beam2` element applies aerodynamic forces to the nodes of a two node structural beam element.

The aerodynamic forces/moment acting on each node are computed based on the relative velocity of a set of locations along the beam and the airstream. The kinematic quantities of the beam are computed based on an interpolation of the kinematics of the three nodes.

Kinematics Interpolation. Note: this part is common to all elements that use the two-node beam discretization and interpolation model. The generic field variable $\mathbf{p}(x)$ is interpolated using linear functions related to the value of the field variable at two locations that in general can be offset from the nodes,

$$\mathbf{p}(\xi) = \sum_{i=1,2} N_i(\xi) \mathbf{p}_i, \quad (12.46)$$

with

$$N_1 = \frac{1}{2} (1 - \xi) \quad (12.47a)$$

$$N_2 = \frac{1}{2} (1 + \xi). \quad (12.47b)$$

Orientation at an arbitrary location ξ :

$$\mathbf{R}(\xi) = \mathbf{R}_{\text{mid}} \exp(\boldsymbol{\theta}(\xi) \times) \quad (12.48)$$

$$\mathbf{R}_{\text{mid}} = \mathbf{R}_{1_a} \exp(\bar{\boldsymbol{\theta}} \times) = \mathbf{R}_{2_a} \exp(\bar{\boldsymbol{\theta}} \times)^T \quad (12.49)$$

$$\bar{\boldsymbol{\theta}} = \frac{1}{2} \text{ax}(\exp^{-1}(\mathbf{R}_{1_a}^T \mathbf{R}_{2_a})) \quad (12.50)$$

$$\boldsymbol{\theta}(\xi) = \sum_{i=1,2} N_i(\xi) \boldsymbol{\theta}_{\text{mid}\leftarrow i} = N_1(\xi) \boldsymbol{\theta}_{\text{mid}\leftarrow 1} + N_2(\xi) \boldsymbol{\theta}_{\text{mid}\leftarrow 2} \quad (12.51)$$

$$\boldsymbol{\theta}_{\text{mid}\leftarrow 1} = -\bar{\boldsymbol{\theta}} \quad (12.52)$$

$$\boldsymbol{\theta}_{\text{mid}\leftarrow 2} = \bar{\boldsymbol{\theta}}. \quad (12.53)$$

Orientation is dealt with specially, given its special nature. The orientation of the mid point is used as a reference, and the orientation parameters that express the relative orientation between each of the end nodes and the mid point are interpolated. The interpolated orientation parameters are used to compute the interpolated relative orientation matrix, which is then pre-multiplied by the orientation matrix of the mid point.

Since the Euler-Rodrigues parametrization is used (the so-called ‘rotation vector’), the magnitude of the relative orientation between each end node must be limited (formally, to π , but it should be less for accuracy).

Note that since the generic orientation $\mathbf{R}(\xi)$ is the result of a sequence of orientations about a common axis $\bar{\boldsymbol{\theta}}$, it can be conveniently rewritten as

$$\begin{aligned} \mathbf{R}(\xi) &= \mathbf{R}_{1_a} \exp(\bar{\boldsymbol{\theta}} \times) \exp(((N_2(\xi) - N_1(\xi)) \bar{\boldsymbol{\theta}}) \times) \\ &= \mathbf{R}_{1_a} \exp(((1 + N_2(\xi) - N_1(\xi)) \bar{\boldsymbol{\theta}}) \times). \end{aligned} \quad (12.54)$$

In fact, when $\xi = -1$, $N_1 = 1$ and $N_2 = 0$, then $\mathbf{R}(\xi) = \mathbf{R}_{1_a}$, while when $\xi = 1$, $N_1 = 0$ and $N_2 = 1$, then $\mathbf{R}(\xi) = \mathbf{R}_{2_a}$.

Position at an arbitrary location ξ :

$$\mathbf{x}(\xi) = \sum_{i=1,2} N_i(\xi) (\mathbf{x}_i + \mathbf{o}_i), \quad (12.55)$$

with $\mathbf{o}_i = \mathbf{R}_i \tilde{\mathbf{o}}_i$.

Angular velocity at an arbitrary location ξ :

$$\boldsymbol{\omega}(\xi) = \sum_{i=1,2} N_i(\xi) \boldsymbol{\omega}_i. \quad (12.56)$$

Velocity at an arbitrary location ξ :

$$\mathbf{v}_{\text{kin}}(\xi) = \sum_{i=1,2} N_i(\xi) (\mathbf{v}_i + \boldsymbol{\omega}_i \times \mathbf{o}_i). \quad (12.57)$$

Perturbation of Interpolated Kinematics. Orientation perturbation at an arbitrary location ξ :

$$\boldsymbol{\theta}_\delta(\xi) = \frac{1 + N_2(\xi) - N_1(\xi)}{2} \boldsymbol{\theta}_{2\delta} + \frac{1 + N_1(\xi) - N_2(\xi)}{2} \boldsymbol{\theta}_{1\delta} \quad (12.58)$$

$$= \sum_{i=1,2} \mathcal{N}_i(\xi) \boldsymbol{\theta}_{i\delta}, \quad (12.59)$$

with

$$\mathcal{N}_i(\xi) = \frac{1 + N_i(\xi) - N_{3-i}(\xi)}{2}. \quad (12.60a)$$

Note, however, that according to Eqs. (12.47), the shape functions of Eq. (12.60a) are $\mathcal{N}_i(\xi) = N_i(\xi)$. Position perturbation at an arbitrary location ξ :

$$\delta\mathbf{x}(\xi) = \sum_{i=1,2} N_i(\xi) (\delta\mathbf{x}_i + \boldsymbol{\theta}_{i\delta} \times \mathbf{o}_i). \quad (12.61)$$

Angular velocity perturbation at an arbitrary location ξ :

$$\begin{aligned} \delta\boldsymbol{\omega}(\xi) &= \sum_{i=1,2} N_i \delta\boldsymbol{\omega}_i \\ &\stackrel{\text{uu}}{=} \sum_{i=1,2} N_i (\delta\dot{\mathbf{g}}_i - \boldsymbol{\omega}_i \times \delta\mathbf{g}_i). \end{aligned} \quad (12.62)$$

Velocity perturbation at an arbitrary location ξ :

$$\begin{aligned} \delta\mathbf{v}_{\text{kin}}(\xi) &= \sum_{i=1,2} N_i(\xi) (\delta\mathbf{v}_i + \delta\boldsymbol{\omega}_i \times \mathbf{o}_i + \boldsymbol{\omega}_i \times \delta\mathbf{o}_i) \\ &= \sum_{i=1,2} N_i(\xi) (\delta\mathbf{v}_i - \mathbf{o}_i \times \delta\boldsymbol{\omega}_i - \boldsymbol{\omega}_i \times \mathbf{o}_i \times \boldsymbol{\theta}_{i\delta}) \\ &\stackrel{\text{uu}}{=} \sum_{i=1,2} N_i(\xi) (\delta\dot{\mathbf{x}}_i - \mathbf{o}_i \times \delta\dot{\mathbf{g}}_i - (\boldsymbol{\omega}_i \times \mathbf{o}_i) \times \delta\mathbf{g}_i). \end{aligned} \quad (12.63)$$

Boundary Conditions Perturbation. Angular velocity perturbation:

$$\begin{aligned} \delta\tilde{\boldsymbol{\omega}}(\xi) &= \mathbf{R}^T(\xi) \sum_{i=1,2} (N_i(\xi) \delta\boldsymbol{\omega}_i + \boldsymbol{\omega}(\xi) \times \mathcal{N}_i(\xi) \boldsymbol{\theta}_{i\delta}) \\ &\stackrel{\text{uu}}{=} \mathbf{R}^T(\xi) \sum_{i=1,2} (N_i(\xi) \delta\dot{\mathbf{g}}_i + (\mathcal{N}_i \boldsymbol{\omega}(\xi) - N_i(\xi) \boldsymbol{\omega}_i) \times \delta\mathbf{g}_i) \\ &= \mathbf{R}^T(\xi) \sum_{i=1,2} N_i(\xi) (\delta\dot{\mathbf{g}}_i + (\boldsymbol{\omega}(\xi) - \boldsymbol{\omega}_i) \times \delta\mathbf{g}_i). \end{aligned} \quad (12.64)$$

Velocity perturbation:

$$\begin{aligned} \delta\tilde{\mathbf{v}}(\xi) &= \mathbf{R}^T(\xi) (N_i(\xi) \delta\mathbf{v}_i - N_i(\xi) \mathbf{o}_i \times \delta\boldsymbol{\omega}_i \\ &\quad + (\mathcal{N}_i(\xi) \mathbf{v}(\xi) \times - N_i(\xi) \boldsymbol{\omega}_i \times \mathbf{o}_i \times) \boldsymbol{\theta}_{i\delta}) \\ &\stackrel{\text{uu}}{=} \mathbf{R}^T(\xi) (N_i(\xi) \delta\dot{\mathbf{x}}_i - N_i(\xi) \mathbf{o}_i \times \delta\dot{\mathbf{g}}_i \\ &\quad + (\mathcal{N}_i(\xi) \mathbf{v}(\xi) - N_i(\xi) \boldsymbol{\omega}_i \times \mathbf{o}_i) \times \delta\mathbf{g}_i) \\ &= \mathbf{R}^T(\xi) N_i(\xi) (\delta\dot{\mathbf{x}}_i - \mathbf{o}_i \times \delta\dot{\mathbf{g}}_i + (\mathbf{v}(\xi) - \boldsymbol{\omega}_i \times \mathbf{o}_i) \times \delta\mathbf{g}_i). \end{aligned} \quad (12.65)$$

They can be summarized as

$$\begin{aligned}
\left\{ \begin{array}{l} \delta\tilde{\boldsymbol{v}}(\xi) \\ \delta\tilde{\boldsymbol{\omega}}(\xi) \end{array} \right\} &= \left[\begin{array}{cc} \boldsymbol{R}^T(\xi) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R}^T(\xi) \end{array} \right] \sum_{i=1,2} \left(N_i(\xi) \left[\begin{array}{cc} \mathbf{I} & -\boldsymbol{o}_i \times \\ \mathbf{0} & \mathbf{I} \end{array} \right] \left\{ \begin{array}{l} \delta\boldsymbol{v}_i \\ \delta\boldsymbol{\omega}_i \end{array} \right\} \right. \\
&\quad \left. + \left[\begin{array}{c} \mathcal{N}_i(\xi) \boldsymbol{v}(\xi) \times -N_i(\xi) \boldsymbol{\omega}_i \times \boldsymbol{o}_i \times \\ \mathcal{N}_i(\xi) \boldsymbol{\omega}(\xi) \times \end{array} \right] \boldsymbol{\theta}_{i\delta} \right) \\
&\stackrel{\text{uu}}{=} \left[\begin{array}{cc} \boldsymbol{R}^T(\xi) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R}^T(\xi) \end{array} \right] \sum_{i=1,2} \left(N_i(\xi) \left[\begin{array}{cc} \mathbf{I} & -\boldsymbol{o}_i \times \\ \mathbf{0} & \mathbf{I} \end{array} \right] \left\{ \begin{array}{l} \delta\dot{\boldsymbol{x}}_i \\ \delta\dot{\boldsymbol{g}}_i \end{array} \right\} \right. \\
&\quad \left. + \left[\begin{array}{c} (\mathcal{N}_i(\xi) \boldsymbol{v}(\xi) - N_i(\xi) \boldsymbol{\omega}_i \times \boldsymbol{o}_i) \times \\ (\mathcal{N}_i(\xi) \boldsymbol{\omega}(\xi) - N_i(\xi) \boldsymbol{\omega}_i) \times \end{array} \right] \delta\boldsymbol{g}_i \right) \\
&= \left[\begin{array}{cc} \boldsymbol{R}^T(\xi) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R}^T(\xi) \end{array} \right] \sum_{i=1,2} N_i(\xi) \left(\left[\begin{array}{cc} \mathbf{I} & -\boldsymbol{o}_i \times \\ \mathbf{0} & \mathbf{I} \end{array} \right] \left\{ \begin{array}{l} \delta\dot{\boldsymbol{x}}_i \\ \delta\dot{\boldsymbol{g}}_i \end{array} \right\} \right. \\
&\quad \left. + \left[\begin{array}{c} (\boldsymbol{v}(\xi) - \boldsymbol{\omega}_i \times \boldsymbol{o}_i) \times \\ (\boldsymbol{\omega}(\xi) - \boldsymbol{\omega}_i) \times \end{array} \right] \delta\boldsymbol{g}_i \right). \tag{12.66}
\end{aligned}$$

i

Contribution to Jacobian Matrix. After defining

$$\boldsymbol{f}_{a/\tilde{\boldsymbol{v}}} = \boldsymbol{R}(\xi) \tilde{\boldsymbol{f}}_{a/\tilde{\boldsymbol{v}}} \boldsymbol{R}^T(\xi) \tag{12.67a}$$

$$\boldsymbol{f}_{a/\tilde{\boldsymbol{\omega}}} = \boldsymbol{R}(\xi) \tilde{\boldsymbol{f}}_{a/\tilde{\boldsymbol{\omega}}} \boldsymbol{R}^T(\xi) \tag{12.67b}$$

$$\boldsymbol{c}_{a/\tilde{\boldsymbol{v}}} = \boldsymbol{R}(\xi) \tilde{\boldsymbol{c}}_{a/\tilde{\boldsymbol{v}}} \boldsymbol{R}^T(\xi) \tag{12.67c}$$

$$\boldsymbol{c}_{a/\tilde{\boldsymbol{\omega}}} = \boldsymbol{R}(\xi) \tilde{\boldsymbol{c}}_{a/\tilde{\boldsymbol{\omega}}} \boldsymbol{R}^T(\xi) \tag{12.67d}$$

$$\begin{aligned}
\boldsymbol{B}_{\tilde{\boldsymbol{v}}} &= (\mathcal{N}_i(\xi) \boldsymbol{v}(\xi) - N_i(\xi) \boldsymbol{\omega}_i \times \boldsymbol{o}_i) \times \\
&= N_i(\xi) (\boldsymbol{v}(\xi) - \boldsymbol{\omega}_i \times \boldsymbol{o}_i) \times \tag{12.67e}
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{B}_{\tilde{\boldsymbol{\omega}}} &= (\mathcal{N}_i(\xi) \boldsymbol{\omega}(\xi) - N_i(\xi) \boldsymbol{\omega}_i) \times \\
&= N_i(\xi) (\boldsymbol{\omega}(\xi) - \boldsymbol{\omega}_i) \times \tag{12.67f}
\end{aligned}$$

$$\boldsymbol{d}_n(\xi) = \boldsymbol{x}(\xi) - \boldsymbol{x}_n, \tag{12.67g}$$

the contribution of the i -th node's motion to the equilibrium of the n -th node ($i = 1, 2, n = 1, 2$) is

$$\begin{aligned}
&N_i(\xi) \left[\begin{array}{c} \boldsymbol{f}_{a/\tilde{\boldsymbol{v}}} \\ \boldsymbol{d}_n(\xi) \times \boldsymbol{f}_{a/\tilde{\boldsymbol{v}}} + \boldsymbol{c}_{a/\tilde{\boldsymbol{v}}} \end{array} \right] \delta\dot{\boldsymbol{x}}_i \\
&+ N_i(\xi) \left[\begin{array}{c} \boldsymbol{f}_{a/\tilde{\boldsymbol{\omega}}} - \boldsymbol{f}_{a/\tilde{\boldsymbol{v}}} \boldsymbol{o}_i \times \\ \boldsymbol{d}_n(\xi) \times \boldsymbol{f}_{a/\tilde{\boldsymbol{\omega}}} + \boldsymbol{c}_{a/\tilde{\boldsymbol{\omega}}} - (\boldsymbol{d}_n(\xi) \times \boldsymbol{f}_{a/\tilde{\boldsymbol{v}}} + \boldsymbol{c}_{a/\tilde{\boldsymbol{v}}}) \boldsymbol{o}_i \times \end{array} \right] \delta\dot{\boldsymbol{g}}_i \\
&+ \left[\begin{array}{c} \boldsymbol{f}_{a/\tilde{\boldsymbol{v}}} \boldsymbol{B}_{\tilde{\boldsymbol{v}}} + \boldsymbol{f}_{a/\tilde{\boldsymbol{\omega}}} \boldsymbol{B}_{\tilde{\boldsymbol{\omega}}} \\ \boldsymbol{d}_n(\xi) \times (\boldsymbol{f}_{a/\tilde{\boldsymbol{v}}} \boldsymbol{B}_{\tilde{\boldsymbol{v}}} + \boldsymbol{f}_{a/\tilde{\boldsymbol{\omega}}} \boldsymbol{B}_{\tilde{\boldsymbol{\omega}}}) + \boldsymbol{c}_{a/\tilde{\boldsymbol{v}}} \boldsymbol{B}_{\tilde{\boldsymbol{v}}} + \boldsymbol{c}_{a/\tilde{\boldsymbol{\omega}}} \boldsymbol{B}_{\tilde{\boldsymbol{\omega}}} \end{array} \right] \delta\boldsymbol{g}_i \\
&\quad + \left[\begin{array}{c} \mathbf{0} \\ -(N_i - \delta_{ni}) \Delta \boldsymbol{f}_n \times \end{array} \right] \delta\boldsymbol{x}_i \\
&+ \left[\begin{array}{c} -\mathcal{N}_i(\xi) \Delta \boldsymbol{f}_n \times \\ -\mathcal{N}_i(\xi) (\boldsymbol{d}_n(\xi) \times \Delta \boldsymbol{f}_n \times + \Delta \boldsymbol{c}_n \times) + \Delta \boldsymbol{f}_n \times N_i \boldsymbol{o}_i \times \end{array} \right] \delta\boldsymbol{g}_i \\
&\stackrel{\text{uu}}{=} \left\{ \begin{array}{l} \delta\Delta\boldsymbol{f}_n \\ \delta\Delta\boldsymbol{c}_n \end{array} \right\}, \tag{12.68}
\end{aligned}$$

where δ_{ni} is Dirac's delta, which is 1 when $i = n$, and 0 otherwise.

Table 12.1: Coefficients of the Wagner indicial response approximation of Theodorsen's function ([1, 2])

	approx. 1	approx. 2
A_1	0.165	0.165
A_2	0.335	0.335
b_1	0.0455	0.041
b_2	0.3	0.32

12.7 Unsteady aerodynamics model

(Author: Mattia Mattaboni)

The 2D unsteady aerodynamics loads are computed implementing the Theodorsen theory in state-space form using the Wagner approximation of the Theodorsen function.

$$\tilde{f}_a = \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c \tilde{\mathbf{c}}_{f_a} (\mathbf{y}(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}, \mathbf{q}), U_\infty) \quad (12.69a)$$

$$\tilde{\mathbf{c}}_a = \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c^2 \tilde{\mathbf{c}}_{c_a} (\mathbf{y}(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}, \mathbf{q}), U_\infty) \quad (12.69b)$$

$$\mathbf{0} = \mathbf{g}(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}, \mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{q}} - \mathbf{A}(U_\infty) \mathbf{q} - \mathbf{B}(U_\infty) \mathbf{u}(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}), \quad (12.69c)$$

where ρ is the air density, c the airfoil chord and \mathbf{y} is

$$\mathbf{y} = \mathbf{C}(U_\infty) \mathbf{q} + \mathbf{D}(U_\infty) \mathbf{u}(\tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}}). \quad (12.70)$$

Matrix \mathbf{A} is

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -b_1 b_2 \left(\frac{2U_\infty}{c} \right)^2 & -(b_1 + b_2) \left(\frac{2U_\infty}{c} \right) & 0 & 0 & 0 & 0 \\ 0 & 0 & -2\omega_{PD} & -\omega_{PD}^2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2\omega_{PD} & -\omega_{PD}^2 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad (12.71)$$

where A_1 , A_2 , b_1 and b_2 are the coefficients of the Theodorsen function approximation (Table 12.1 from [1] and [2]), ω_{PD} is the frequency of the pseudo-derivative algorithm

$$\mathcal{L}(\dot{f}) = s\mathcal{L}(f) - f(0) \cong \frac{s}{(1 + s/\omega_{PD})^2} \mathcal{L}(f) \quad (12.72)$$

(where $f(0)$ can be neglected under broad assumptions), and

$$U_\infty = \sqrt{\tilde{v}_x^2 + \tilde{v}_y^2}. \quad (12.73)$$

Matrix \mathbf{B} is

$$\mathbf{B} = \tilde{\mathbf{B}} \mathbf{T}_u(U_\infty), \quad (12.74)$$

where

$$\tilde{\mathbf{B}} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad (12.75)$$

and

$$\mathbf{T}_u = \left[\begin{array}{cc} 0 & 1 \\ 1 & -\frac{d_{1/4}}{U_\infty} \\ 1 & -\frac{d_{3/4}}{U_\infty} \end{array} \right]^{-1} = \left[\begin{array}{c} 0 & 1 \\ \frac{1}{d_{3/4}-d_{1/4}} \begin{bmatrix} d_{3/4} & -d_{1/4} \\ U_\infty & -U_\infty \end{bmatrix} \end{array} \right]. \quad (12.76)$$

$$\mathbf{B} = \left[\begin{array}{cc} 0 & 0 \\ 0 & 1 \\ \frac{1}{d_{3/4}-d_{1/4}} d_{3/4} & -\frac{1}{d_{3/4}-d_{1/4}} d_{1/4} \\ 0 & 0 \\ \frac{1}{d_{3/4}-d_{1/4}} U_\infty & -\frac{1}{d_{3/4}-d_{1/4}} U_\infty \\ 0 & 0 \end{array} \right] \quad (12.77)$$

$$\mathbf{u} = \left\{ \begin{array}{l} \tan^{-1} \left(\frac{-V_y - \omega_z d_{1/4}}{V_x} \right) \\ \tan^{-1} \left(\frac{-V_y - \omega_z d_{3/4}}{V_x} \right) \end{array} \right\} \quad (12.78)$$

Matrix \mathbf{C} is

$$\mathbf{C} = \left[\begin{array}{cccccc} (A_1 + A_2) b_1 b_2 \left(\frac{2U_\infty}{c} \right)^2 & (A_1 b_1 + A_2 b_2) \left(\frac{2U_\infty}{c} \right) & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_{PD}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega_{PD}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right]. \quad (12.79)$$

Matrix \mathbf{D} is

$$\mathbf{D} = \tilde{\mathbf{D}} \mathbf{T}_u (U_\infty), \quad (12.80)$$

where

$$\tilde{\mathbf{D}} = \left[\begin{array}{ccc} (1 - A_1 - A_2) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{array} \right]. \quad (12.81)$$

$$\mathbf{D} = \left[\begin{array}{cc} 0 & (1 - A_1 - A_2) \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{d_{3/4}-d_{1/4}} U_\infty & -\frac{1}{d_{3/4}-d_{1/4}} U_\infty \end{array} \right]. \quad (12.82)$$

$$\tilde{\mathbf{c}}_{f_a} = \left\{ \begin{array}{c} -c_d \\ c_l \\ 0 \end{array} \right\} = \mathbf{c}_{f_a}^{\text{lookup}} (\mathbf{y}) + \tilde{\mathbf{T}}_{f_a} \mathbf{y}. \quad (12.83)$$

$$\tilde{\mathbf{c}}_{c_a} = \left\{ \begin{array}{c} 0 \\ 0 \\ c_m \end{array} \right\} = \mathbf{c}_{c_a}^{\text{lookup}} (\mathbf{y}) + \tilde{\mathbf{T}}_{c_a} \mathbf{y}. \quad (12.84)$$

The second part is the noncirculatory effect.

$$\begin{Bmatrix} c_l^{NC} \\ c_m^{NC} \end{Bmatrix} = \mathbf{T}_1 \mathbf{T}_2 (U_\infty) \mathbf{y} = \tilde{\mathbf{T}} \mathbf{y}, \quad (12.85)$$

where

$$\mathbf{T}_1 = \begin{bmatrix} 1 & 0 \\ -\frac{a+\frac{1}{2}}{2} & 1 \end{bmatrix}, \quad (12.86)$$

and

$$\mathbf{T}_2 = \frac{C_{l_\alpha}}{2} \frac{c}{2} \begin{bmatrix} 0 & \frac{1}{U_\infty} & -\frac{ca}{2U_\infty^2} & 0 \\ 0 & \frac{a}{2U_\infty} & -\frac{c}{4U_\infty^2} \left(\frac{1}{8} + a^2 \right) & -\frac{1}{4U_\infty} \end{bmatrix}. \quad (12.87)$$

$$\tilde{\mathbf{T}} = \frac{C_{l_\alpha}}{2} \frac{c}{2} \begin{bmatrix} 0 & \frac{1}{U_\infty} & -\frac{ca}{2U_\infty^2} & 0 \\ 0 & -\frac{1}{4U_\infty} & \left(\frac{ca}{8U_\infty^2} - \frac{c}{32U_\infty^2} \right) & -\frac{1}{4U_\infty} \end{bmatrix} \quad (12.88)$$

$$\tilde{\mathbf{T}}_{f_a} = \frac{C_{l_\alpha}}{2} \frac{c}{2} \begin{bmatrix} 0 & \frac{1}{U_\infty} & -\frac{ca}{2U_\infty^2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (12.89)$$

$$\tilde{\mathbf{T}}_{c_a} = \frac{C_{l_\alpha}}{2} \frac{c}{2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{4U_\infty} & \left(\frac{ca}{8U_\infty^2} - \frac{c}{32U_\infty^2} \right) & -\frac{1}{4U_\infty} \end{bmatrix} \quad (12.90)$$

Finally,

$$\tilde{\mathbf{c}}_{f_a} = \mathbf{c}_{f_a}^{\text{lookup}} (y_1) + \begin{Bmatrix} 0 \\ \frac{c_{l_\alpha}}{2} \frac{c}{2U_\infty} \left(y_2 - \frac{ca}{2U_\infty} y_3 \right) \\ 0 \end{Bmatrix}. \quad (12.91)$$

$$\tilde{\mathbf{c}}_{c_a} = \mathbf{c}_{c_a}^{\text{lookup}} (y_1) + \begin{Bmatrix} 0 \\ 0 \\ \frac{c_{l_\alpha}}{2} \frac{c}{2U_\infty} \left(-\frac{1}{4} y_2 + \left(\frac{ca}{8U_\infty} - \frac{c}{32U_\infty} \right) y_3 - \frac{1}{4} y_4 \right) \end{Bmatrix}. \quad (12.92)$$

12.7.1 Perturbation of the Equations

Perturbation of \mathbf{g} with respect to $\tilde{\mathbf{v}}$

The perturbation of \mathbf{g} is

$$\mathbf{g}/\tilde{\mathbf{v}} = - (\mathbf{A}_{/U_\infty} \mathbf{q} + \mathbf{B}_{/U_\infty} \mathbf{u}) U_\infty/\tilde{\mathbf{v}} - \mathbf{B}\mathbf{u}/\tilde{\mathbf{v}}. \quad (12.93)$$

where, according to the previous definition of the matrices, the derivatives are

$$\mathbf{A}_{/U_\infty} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -b_1 b_2 \frac{8}{c^2} U_\infty & -(b_1 + b_2) \frac{2}{c} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (12.94)$$

$$\mathbf{B}_{/U_\infty} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{d_{3/4}-d_{1/4}} & -\frac{1}{d_{3/4}-d_{1/4}} \\ 0 & 0 \end{bmatrix} \quad (12.95)$$

$$U_{\infty/\tilde{\mathbf{v}}} = \begin{bmatrix} \frac{\tilde{v}_x}{U_\infty} & \frac{\tilde{v}_y}{U_\infty} & 0 \end{bmatrix} \quad (12.96)$$

$$\mathbf{u}_{/\tilde{\mathbf{v}}} = \begin{bmatrix} \frac{\tilde{v}_y + \tilde{\omega}_z d_{1/4}}{\tilde{v}_x^2 + \tilde{v}_y^2 + \tilde{\omega}_z^2 d_{1/4}^2 + 2\tilde{v}_y \tilde{\omega}_z d_{1/4}} & \frac{-\tilde{v}_x}{\tilde{v}_x^2 + \tilde{v}_y^2 + \tilde{\omega}_z^2 d_{1/4}^2 + 2\tilde{v}_y \tilde{\omega}_z d_{1/4}} & 0 \\ \frac{\tilde{v}_y + \tilde{\omega}_z d_{3/4}}{\tilde{v}_x^2 + \tilde{v}_y^2 + \tilde{\omega}_z^2 d_{3/4}^2 + 2\tilde{v}_y \tilde{\omega}_z d_{3/4}} & \frac{-\tilde{v}_x}{\tilde{v}_x^2 + \tilde{v}_y^2 + \tilde{\omega}_z^2 d_{3/4}^2 + 2\tilde{v}_y \tilde{\omega}_z d_{3/4}} & 0 \end{bmatrix} \quad (12.97)$$

The explicit computation of each term of Eq. 12.93 yields

$$(\mathbf{A}_{U_\infty} \mathbf{q} + \mathbf{B}_{U_\infty} \mathbf{u}) U_{\infty/\tilde{\mathbf{v}}} = \left\{ \begin{array}{c} 0 \\ -b_1 b_2 \frac{8}{c^2} U_\infty q_1 - (b_1 + b_2) \frac{2}{c} q_2 \\ 0 \\ 0 \\ \frac{u_1 - u_2}{d_{3/4} - d_{1/4}} \\ 0 \end{array} \right\} \left\{ \begin{array}{c} \frac{\tilde{v}_x}{U_\infty} \\ \frac{\tilde{v}_x}{U_\infty} \\ 0 \end{array} \right\} \quad (12.98)$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ (-b_1 b_2 \frac{8}{c^2} U_\infty q_1 - (b_1 + b_2) \frac{2}{c} q_2) \frac{\tilde{v}_x}{U_\infty} & (-b_1 b_2 \frac{8}{c^2} U_\infty q_1 - (b_1 + b_2) \frac{2}{c} q_2) \frac{\tilde{v}_y}{U_\infty} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \left(\frac{u_1 - u_2}{d_{3/4} - d_{1/4}} \right) \frac{\tilde{v}_x}{U_\infty} & \left(\frac{u_1 - u_2}{d_{3/4} - d_{1/4}} \right) \frac{\tilde{v}_y}{U_\infty} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (12.99)$$

for the first, while the second yields

$$\mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}} = \begin{bmatrix} 0 & 0 & 0 \\ \mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}}(2,1) & \mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}}(2,2) & 0 \\ \mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}}(3,1) & \mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}}(3,2) & 0 \\ 0 & 0 & 0 \\ \mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}}(5,1) & \mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}}(5,2) & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (12.100)$$

The non-null terms are

$$\mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}}(2,1) = \mathbf{u}_{/\tilde{\mathbf{v}}}(2,1) \quad (12.101)$$

$$\mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}}(2,2) = \mathbf{u}_{/\tilde{\mathbf{v}}}(2,2) \quad (12.102)$$

$$\mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}}(3,1) = \mathbf{B}(3,1)\mathbf{u}_{/\tilde{\mathbf{v}}}(1,1) + \mathbf{B}(3,2)\mathbf{u}_{/\tilde{\mathbf{v}}}(2,1) \quad (12.103)$$

$$\mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}}(3,2) = \mathbf{B}(3,1)\mathbf{u}_{/\tilde{\mathbf{v}}}(1,2) + \mathbf{B}(3,2)\mathbf{u}_{/\tilde{\mathbf{v}}}(2,2) \quad (12.104)$$

$$\mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}}(3,1) = \mathbf{B}(5,1)\mathbf{u}_{/\tilde{\mathbf{v}}}(1,1) + \mathbf{B}(5,2)\mathbf{u}_{/\tilde{\mathbf{v}}}(2,1) \quad (12.105)$$

$$\mathbf{B}\mathbf{u}_{/\tilde{\mathbf{v}}}(3,2) = \mathbf{B}(5,1)\mathbf{u}_{/\tilde{\mathbf{v}}}(1,2) + \mathbf{B}(5,2)\mathbf{u}_{/\tilde{\mathbf{v}}}(2,2) \quad (12.106)$$

So, the Jacobian matrix $\mathbf{g}_{/\tilde{\mathbf{v}}}$ is

$$\mathbf{g}_{/\tilde{\mathbf{v}}} = \begin{bmatrix} 0 & 0 & 0 \\ \mathbf{g}_{/\tilde{\mathbf{v}}}(2,1) & \mathbf{g}_{/\tilde{\mathbf{v}}}(2,2) & 0 \\ \mathbf{g}_{/\tilde{\mathbf{v}}}(3,1) & \mathbf{g}_{/\tilde{\mathbf{v}}}(3,2) & 0 \\ 0 & 0 & 0 \\ \mathbf{g}_{/\tilde{\mathbf{v}}}(5,1) & \mathbf{g}_{/\tilde{\mathbf{v}}}(5,2) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (12.107)$$

where the matrix elements are

$$\mathbf{g}_{/\tilde{\mathbf{v}}}(2,1) = -\left(-b_1 b_2 \frac{8}{c^2} U_\infty q_1 - (b_1 + b_2) \frac{2}{c} q_2\right) \frac{\tilde{v}_x}{U_\infty} - \mathbf{u}_{/\tilde{\mathbf{v}}}(2,1) \quad (12.108)$$

$$\mathbf{g}_{/\tilde{\mathbf{v}}}(2,2) = -\left(-b_1 b_2 \frac{8}{c^2} U_\infty q_1 - (b_1 + b_2) \frac{2}{c} q_2\right) \frac{\tilde{v}_y}{U_\infty} - \mathbf{u}_{/\tilde{\mathbf{v}}}(2,2) \quad (12.109)$$

$$\mathbf{g}_{/\tilde{\mathbf{v}}}(3,1) = -(\mathbf{B}(3,1)\mathbf{u}_{/\tilde{\mathbf{v}}}(1,1) + \mathbf{B}(3,2)\mathbf{u}_{/\tilde{\mathbf{v}}}(2,1)) \quad (12.110)$$

$$\mathbf{g}_{/\tilde{\mathbf{v}}}(3,2) = -(\mathbf{B}(3,1)\mathbf{u}_{/\tilde{\mathbf{v}}}(1,2) + \mathbf{B}(3,2)\mathbf{u}_{/\tilde{\mathbf{v}}}(2,2)) \quad (12.111)$$

$$\mathbf{g}_{\tilde{\mathbf{v}}}(3,1) = -\left(\frac{u_1 - u_2}{d_{3/4} - d_{1/4}}\right) \frac{\tilde{v}_x}{U_\infty} - (\mathbf{B}(5,1)\mathbf{u}_{/\tilde{\mathbf{v}}}(1,1) + \mathbf{B}(5,2)\mathbf{u}_{/\tilde{\mathbf{v}}}(2,1)) \quad (12.112)$$

$$\mathbf{g}_{\tilde{\mathbf{v}}}(3,2) = -\left(\frac{u_1 - u_2}{d_{3/4} - d_{1/4}}\right) \frac{\tilde{v}_y}{U_\infty} - (\mathbf{B}(5,1)\mathbf{u}_{/\tilde{\mathbf{v}}}(1,2) + \mathbf{B}(5,2)\mathbf{u}_{/\tilde{\mathbf{v}}}(2,2)) \quad (12.113)$$

Perturbation of \mathbf{g} with respect to $\tilde{\omega}$

The perturbation of \mathbf{g} yields

$$\mathbf{g}_{/\tilde{\omega}} = -\mathbf{B}\mathbf{u}_{/\tilde{\omega}}. \quad (12.114)$$

Where

$$\mathbf{u}_{/\tilde{\omega}} = \begin{bmatrix} 0 & 0 & \frac{-\tilde{v}_x d_{1/4}}{\tilde{v}_x^2 + \tilde{v}_y^2 + \tilde{\omega}_z^2 d_{1/4}^2 + 2\tilde{v}_y \tilde{\omega}_z d_{1/4}} \\ 0 & 0 & \frac{-\tilde{v}_x d_{3/4}}{\tilde{v}_x^2 + \tilde{v}_y^2 + \tilde{\omega}_z^2 d_{3/4}^2 + 2\tilde{v}_y \tilde{\omega}_z d_{3/4}} \end{bmatrix} \quad (12.115)$$

Thus,

$$\mathbf{g}_{/\tilde{\omega}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \mathbf{g}_{/\tilde{\omega}}(2,3) \\ 0 & 0 & \mathbf{g}_{/\tilde{\omega}}(3,3) \\ 0 & 0 & 0 \\ 0 & 0 & \mathbf{g}_{/\tilde{\omega}}(5,3) \\ 0 & 0 & 0 \end{bmatrix} \quad (12.116)$$

where

$$\mathbf{g}_{/\tilde{\omega}}(2,3) = -\mathbf{u}_{/\tilde{\omega}}(2,3) \quad (12.117)$$

$$\mathbf{g}_{/\tilde{\omega}}(5,3) = -(\mathbf{B}(3,1)\mathbf{u}_{/\tilde{\omega}}(1,3) + \mathbf{B}(3,2)\mathbf{u}_{/\tilde{\omega}}(2,3)) \quad (12.118)$$

$$\mathbf{g}_{/\tilde{\omega}}(5,3) = -(\mathbf{B}(5,1)\mathbf{u}_{/\tilde{\omega}}(1,3) + \mathbf{B}(5,2)\mathbf{u}_{/\tilde{\omega}}(2,3)) \quad (12.119)$$

Perturbation of \mathbf{g} with respect to \mathbf{q}

The perturbation of \mathbf{g} yields

$$\mathbf{g}_{/\mathbf{q}} = -\mathbf{A} \quad (12.120)$$

Perturbation of \mathbf{g} with respect to $\dot{\mathbf{q}}$

The perturbation of \mathbf{g} yields

$$\mathbf{g}_{/\dot{\mathbf{q}}} = \mathbf{I} \quad (12.121)$$

12.7.2 Perturbation of the aerodynamic forces

Perturbation of $\tilde{\mathbf{f}}_a$ with respect to $\tilde{\mathbf{v}}$

$$\tilde{\mathbf{f}}_{a/\tilde{\mathbf{v}}} = \rho c \tilde{\mathbf{c}}_{f_a} \begin{bmatrix} \tilde{v}_x & \tilde{v}_y & \tilde{v}_z \end{bmatrix} + \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c \left(\tilde{\mathbf{c}}_{f_a/\mathbf{y}} \mathbf{y}_{/\tilde{\mathbf{v}}} + \tilde{\mathbf{c}}_{f_a/U_\infty} U_{\infty/\tilde{\mathbf{v}}} \right) \quad (12.122)$$

where

$$\mathbf{y}_{/\tilde{\mathbf{v}}} = (\mathbf{C}_{/U_\infty} \mathbf{q} + \mathbf{D}_{/U_\infty} \mathbf{u}) U_{\infty/\tilde{\mathbf{v}}} + \mathbf{D} \mathbf{u}_{/\tilde{\mathbf{v}}} \quad (12.123)$$

$$\mathbf{C}_{/U_\infty} = \begin{bmatrix} (A_1 + A_2) b_1 b_2 \frac{8}{c^2} U_\infty & (A_1 b_1 + A_2 b_2) \frac{2}{c} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (12.124)$$

$$\mathbf{D}_{/U_\infty} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{d_{3/4}-d_{1/4}} & -\frac{1}{d_{3/4}-d_{1/4}} \end{bmatrix} \quad (12.125)$$

$$\tilde{\mathbf{c}}_{f_a/U_\infty} = \left\{ \begin{array}{c} 0 \\ \frac{C_{l_\alpha}}{2} \frac{c}{2} \frac{1}{U_\infty^2} \left(-y_2 + \frac{ca}{U_\infty} y_3 \right) \\ 0 \end{array} \right\} \quad (12.126)$$

$$\tilde{\mathbf{c}}_{f_a/y} = \begin{bmatrix} \frac{dc_d^{\text{lookup}}}{d\alpha} & 0 & 0 & 0 \\ \frac{dc_l^{\text{lookup}}}{d\alpha} & \frac{C_{l_\alpha}}{2} \frac{c}{2U_\infty} & -\frac{c_{l_\alpha}}{2} \frac{c^2 a}{4U_\infty^2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (12.127)$$

Perturbation of \tilde{f}_a with respect to $\tilde{\omega}$

$$\tilde{\mathbf{f}}_{a/\tilde{\omega}} = \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c \left(\tilde{\mathbf{c}}_{f_a/y} \mathbf{y}_{/\tilde{\omega}} \right) \quad (12.128)$$

where

$$\mathbf{y}_{/\tilde{\omega}} = \mathbf{D} \mathbf{u}_{/\tilde{\omega}} \quad (12.129)$$

Perturbation of \tilde{f}_a with respect to q

$$\tilde{\mathbf{f}}_{a/q} = \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c \left(\tilde{\mathbf{c}}_{f_a/y} \mathbf{y}_{/q} \right) \quad (12.130)$$

where

$$\mathbf{y}_{/q} = \mathbf{C} \quad (12.131)$$

Perturbation of \tilde{f}_a with respect to \dot{q}

$$\tilde{\mathbf{f}}_{a/\dot{q}} = 0 \quad (12.132)$$

12.7.3 Perturbation of the aerodynamic moments

Perturbation of \tilde{c}_a with respect to $\tilde{\mathbf{v}}$

$$\tilde{\mathbf{c}}_{a/\tilde{\mathbf{v}}} = \rho c^2 \tilde{\mathbf{c}}_{c_a} \begin{bmatrix} \tilde{v}_x & \tilde{v}_y & \tilde{v}_z \end{bmatrix} + \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c^2 \left(\tilde{\mathbf{c}}_{c_a/y} \mathbf{y}_{/\tilde{\mathbf{v}}} + \tilde{\mathbf{c}}_{c_a/U_\infty} U_\infty/\tilde{\mathbf{v}} \right) \quad (12.133)$$

$$\tilde{\mathbf{c}}_{c_a/U_\infty} = \left\{ \begin{array}{c} 0 \\ 0 \\ \frac{c_{l_\alpha}}{2} \frac{c}{2} \frac{1}{4U_\infty^2} \left(y_2 - \left(\frac{ca}{U_\infty} - \frac{c}{4U_\infty} \right) y_3 + y_4 \right) \end{array} \right\} \quad (12.134)$$

$$\tilde{\mathbf{c}}_{c_a/y} = \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{dc_m^{\text{lookup}}}{d\alpha} & -\frac{1}{4} \frac{c_{l_\alpha}}{2} \frac{c}{2U_\infty} & \frac{c_{l_\alpha}}{2} \frac{c}{2U_\infty} \left(\frac{ca}{8U_\infty} - \frac{c}{32U_\infty} \right) & -\frac{1}{4} \frac{c_{l_\alpha}}{2} \frac{c}{2U_\infty} \end{array} \right] \quad (12.135)$$

Perturbation of \tilde{c}_a with respect to $\tilde{\omega}$

$$\tilde{c}_{a/\tilde{\omega}} = \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c^2 \left(\tilde{\mathbf{c}}_{c_a/y} \mathbf{y}/\tilde{\omega} \right) \quad (12.136)$$

Perturbation of \tilde{c}_a with respect to q

$$\tilde{c}_{a/q} = \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c^2 \left(\tilde{\mathbf{c}}_{c_a/y} \mathbf{y}/q \right) \quad (12.137)$$

Perturbation of \tilde{c}_a with respect to \dot{q}

$$\tilde{c}_{a/\dot{q}} = 0 \quad (12.138)$$

12.7.4 Finite difference version

A reduced state-space model, with just 2 states instead of 6 states, can be used deleting the pseudo-derivative algorithm and computing the derivate using the backward finite difference.

The set of equations that describes the unsteady aerodynamic loads is still the same:

$$\tilde{\mathbf{f}}_a = \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c \tilde{\mathbf{c}}_{f_a} (y(\tilde{\mathbf{v}}, \tilde{\omega}, q), U_\infty) \quad (12.139a)$$

$$\tilde{c}_a = \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c^2 \tilde{\mathbf{c}}_{c_a} (y(\tilde{\mathbf{v}}, \tilde{\omega}, q), U_\infty) \quad (12.139b)$$

$$\mathbf{0} = \mathbf{g}(\tilde{\mathbf{v}}, \tilde{\omega}, q, \dot{q}) = \dot{q} - \mathbf{A}(U_\infty) q - \mathbf{B} u(\tilde{\mathbf{v}}, \tilde{\omega}), \quad (12.139c)$$

where ρ is the air density, c the airfoil chord and \mathbf{y} is defined as:

$$y = \mathbf{C}(U_\infty) q + \mathbf{D} u(\tilde{\mathbf{v}}, \tilde{\omega}). \quad (12.140)$$

In this case the matrix \mathbf{A} is:

$$\mathbf{A} = \left[\begin{array}{cc} 0 & 1 \\ -b_1 b_2 \left(\frac{2U_\infty}{c} \right)^2 & -(b_1 + b_2) \left(\frac{2U_\infty}{c} \right) \end{array} \right], \quad (12.141)$$

where A_1 , A_2 , b_1 and b_2 are the coefficients of the Theodorsen function approximation (Table 12.1 from [1] and [2]),

The matrix \mathbf{B} is simply:

$$\mathbf{B} = \left[\begin{array}{c} 0 \\ 1 \end{array} \right], \quad (12.142)$$

Whereas the input of the system u is the angle of attack computed at the 3/4 chord point:

$$u = \tan^{-1} \left(\frac{-V_y - \omega_z d_{3/4}}{V_x} \right) \quad (12.143)$$

The \mathbf{C} matrix is:

$$\mathbf{C} = \begin{bmatrix} (A_1 + A_2) b_1 b_2 \left(\frac{2U_\infty}{c} \right)^2 & (A_1 b_1 + A_2 b_2) \left(\frac{2U_\infty}{c} \right) \end{bmatrix}. \quad (12.144)$$

and the \mathbf{D} matrix:

$$\mathbf{D} = \begin{bmatrix} (1 - A_1 - A_2) \end{bmatrix}. \quad (12.145)$$

Using the output \mathbf{y} of this SISO state-space model as input for the lookup table the circulatory part of the unsteady loads is computed.

The non-circulatory part is computed starting from the angle of attack computed at 1/4 chord (u_1) and 3/4 chord (u_2):

$$\begin{Bmatrix} \alpha_{pivot} \\ \dot{\alpha} \end{Bmatrix} = \begin{bmatrix} 1 & -\frac{d_{1/4}}{U_\infty} \\ 1 & -\frac{d_{3/4}}{U_\infty} \end{bmatrix}^{-1} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \frac{1}{d_{3/4} - d_{1/4}} \begin{bmatrix} d_{3/4} & -d_{1/4} \\ U_\infty & -U_\infty \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \quad (12.146)$$

where:

$$\mathbf{u} = \begin{Bmatrix} \tan^{-1} \left(\frac{-V_y - \omega_z d_{1/4}}{V_x} \right) \\ \tan^{-1} \left(\frac{-V_y - \omega_z d_{3/4}}{V_x} \right) \end{Bmatrix} \quad (12.147)$$

Thus, the aerodynamic coefficients result:

$$\tilde{\mathbf{c}}_{f_a} = \mathbf{c}_{f_a}^{\text{lookup}}(y) + \begin{Bmatrix} 0 \\ \frac{c_{l\alpha}}{2} \frac{c}{2U_\infty} \left(\dot{\alpha}_{pivot} - \frac{ca}{2U_\infty} \ddot{\alpha} \right) \\ 0 \end{Bmatrix}. \quad (12.148)$$

$$\tilde{\mathbf{c}}_{c_a} = \mathbf{c}_{c_a}^{\text{lookup}}(y) + \begin{Bmatrix} 0 \\ 0 \\ \frac{c_{l\alpha}}{2} \frac{c}{2U_\infty} \left(-\frac{1}{4} \dot{\alpha}_{pivot} + \left(\frac{ca}{8U_\infty} - \frac{c}{32U_\infty} \right) \ddot{\alpha} - \frac{1}{4} \dot{\alpha} \right) \end{Bmatrix}. \quad (12.149)$$

In order to compute the aerodynamic coefficients $\dot{\alpha}_{pivot}$ and $\ddot{\alpha}$ are necessary and they can be computed using the backward finite difference:

$$\begin{Bmatrix} \dot{\alpha}_{pivot} \\ \ddot{\alpha} \end{Bmatrix} = \frac{1}{\Delta t} \left(\begin{Bmatrix} \alpha_{pivot} \\ \dot{\alpha} \end{Bmatrix}_k - \begin{Bmatrix} \alpha_{pivot} \\ \dot{\alpha} \end{Bmatrix}_{k-1} \right) \quad (12.150)$$

12.7.5 Perturbation of the equations

Perturbation of \mathbf{g} with respect to $\tilde{\mathbf{v}}$

The perturbation of \mathbf{g} is:

$$\mathbf{g}/\tilde{\mathbf{v}} = -\mathbf{A}_{/U_\infty} \mathbf{q} U_\infty/\tilde{\mathbf{v}} - \mathbf{B} u/\tilde{\mathbf{v}}. \quad (12.151)$$

where, accordingly with the previous definition of the matrices the derivatives are:

$$\mathbf{A}_{U_\infty} = \begin{bmatrix} 0 & 0 \\ -b_1 b_2 \frac{8}{c^2} U_\infty & -(b_1 + b_2) \frac{2}{c} \end{bmatrix} \quad (12.152)$$

$$U_{\infty/\tilde{\mathbf{v}}} = \begin{bmatrix} \frac{\tilde{v}_x}{U_\infty} & \frac{\tilde{v}_y}{U_\infty} & 0 \end{bmatrix} \quad (12.153)$$

$$u_{/\tilde{\mathbf{v}}} = \begin{bmatrix} \frac{\tilde{v}_y + \tilde{\omega}_z d_{3/4}}{\tilde{v}_x^2 + \tilde{v}_y^2 + \tilde{\omega}_z^2 d_{3/4}^2 + 2\tilde{v}_y \tilde{\omega}_z d_{3/4}} & \frac{-\tilde{v}_x}{\tilde{v}_x^2 + \tilde{v}_y^2 + \tilde{\omega}_z^2 d_{3/4}^2 + 2\tilde{v}_y \tilde{\omega}_z d_{3/4}} & 0 \end{bmatrix} \quad (12.154)$$

We can now explicitly perform the computation of each term of Eq. 12.151:

$$\mathbf{A}_{U_\infty} \mathbf{q} U_{\infty/\tilde{\mathbf{v}}} = \left\{ \begin{array}{c} 0 \\ -b_1 b_2 \frac{8}{c^2} U_\infty q_1 - (b_1 + b_2) \frac{2}{c} q_2 \end{array} \right\} \left\{ \begin{array}{c} \frac{\tilde{v}_x}{U_\infty} \\ \frac{\tilde{v}_y}{U_\infty} \\ 0 \end{array} \right\} \quad (12.155)$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ (-b_1 b_2 \frac{8}{c^2} U_\infty q_1 - (b_1 + b_2) \frac{2}{c} q_2) \frac{\tilde{v}_x}{U_\infty} & (-b_1 b_2 \frac{8}{c^2} U_\infty q_1 - (b_1 + b_2) \frac{2}{c} q_2) \frac{\tilde{v}_y}{U_\infty} & 0 \end{bmatrix} \quad (12.156)$$

wheres the second is:

$$\mathbf{B} u_{/\tilde{\mathbf{v}}} = \begin{bmatrix} 0 & 0 & 0 \\ u_{/\tilde{\mathbf{v}}}(1,1) & u_{/\tilde{\mathbf{v}}}(1,2) & 0 \end{bmatrix} \quad (12.157)$$

So, the jacobian matrix $\mathbf{g}_{/\tilde{\mathbf{v}}}$ results:

$$\mathbf{g}_{/\tilde{\mathbf{v}}} = \begin{bmatrix} 0 & 0 & 0 \\ \mathbf{g}_{/\tilde{\mathbf{v}}}(2,1) & \mathbf{g}_{/\tilde{\mathbf{v}}}(2,2) & 0 \end{bmatrix} \quad (12.158)$$

where the matrix elements are:

$$\mathbf{g}_{/\tilde{\mathbf{v}}}(2,1) = - \left(-b_1 b_2 \frac{8}{c^2} U_\infty q_1 - (b_1 + b_2) \frac{2}{c} q_2 \right) \frac{\tilde{v}_x}{U_\infty} - u_{/\tilde{\mathbf{v}}}(1,1) \quad (12.159)$$

$$\mathbf{g}_{/\tilde{\mathbf{v}}}(2,2) = - \left(-b_1 b_2 \frac{8}{c^2} U_\infty q_1 - (b_1 + b_2) \frac{2}{c} q_2 \right) \frac{\tilde{v}_y}{U_\infty} - u_{/\tilde{\mathbf{v}}}(1,2) \quad (12.160)$$

Perturbation of \mathbf{g} with respect to $\tilde{\omega}$

The perturbation of \mathbf{g} is:

$$\mathbf{g}_{/\tilde{\omega}} = -\mathbf{B} u_{/\tilde{\omega}}. \quad (12.161)$$

Where:

$$u_{/\tilde{\omega}} = \begin{bmatrix} 0 & 0 & \frac{-\tilde{v}_x d_{3/4}}{\tilde{v}_x^2 + \tilde{v}_y^2 + \tilde{\omega}_z^2 d_{3/4}^2 + 2\tilde{v}_y \tilde{\omega}_z d_{3/4}} \end{bmatrix} \quad (12.162)$$

Thus, it results:

$$\mathbf{g}_{/\tilde{\omega}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -u_{/\tilde{\omega}}(1,3) \end{bmatrix} \quad (12.163)$$

Perturbation of \mathbf{g} with respect to \mathbf{q}

The perturbation of \mathbf{g} is simply:

$$\mathbf{g}_{/\mathbf{q}} = -\mathbf{A} \quad (12.164)$$

Perturbation of \mathbf{g} with respect to $\dot{\mathbf{q}}$

The perturbation of \mathbf{g} is simply:

$$\mathbf{g}_{/\dot{\mathbf{q}}} = \mathbf{I} \quad (12.165)$$

12.7.6 Perturbation of the aerodynamic forces

Perturbation of $\tilde{\mathbf{f}}_a$ with respect to $\tilde{\mathbf{v}}$

$$\tilde{\mathbf{f}}_{a/\tilde{\mathbf{v}}} = \rho c \tilde{\mathbf{c}}_{f_a} \begin{bmatrix} \tilde{v}_x & \tilde{v}_y & \tilde{v}_z \end{bmatrix} + \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c \left(\tilde{\mathbf{c}}_{f_{a/y}} y_{/\tilde{\mathbf{v}}} + \tilde{\mathbf{c}}_{f_{a/U_\infty}} U_\infty_{/\tilde{\mathbf{v}}} \right) \quad (12.166)$$

where:

$$y_{/\tilde{\mathbf{v}}} = \mathbf{C}_{/U_\infty} \mathbf{q} U_\infty_{/\tilde{\mathbf{v}}} + \mathbf{D} u_{/\tilde{\mathbf{v}}} \quad (12.167)$$

$$\mathbf{C}_{/U_\infty} = \begin{bmatrix} (A_1 + A_2) b_1 b_2 \frac{8}{c^2} U_\infty & (A_1 b_1 + A_2 b_2) \frac{2}{c} \end{bmatrix} \quad (12.168)$$

$$\tilde{\mathbf{c}}_{f_{a/U_\infty}} = \begin{Bmatrix} 0 \\ \frac{C_{l_\alpha}}{2} \frac{c}{2} \frac{1}{U_\infty^2} \left(-\dot{\alpha}_{pivot} + \frac{ca}{U_\infty} \ddot{\alpha} \right) \\ 0 \end{Bmatrix} \quad (12.169)$$

$$\tilde{\mathbf{c}}_{f_{a/y}} = \begin{Bmatrix} \frac{dc_d^{\text{lookup}}}{d\alpha} \\ \frac{dc_l^{\text{lookup}}}{d\alpha} \\ 0 \end{Bmatrix} \quad (12.170)$$

where the dependence of $\dot{\alpha}_{pivot}$, $\dot{\alpha}$ and $\ddot{\alpha}$ is neglected.

Perturbation of $\tilde{\mathbf{f}}_a$ with respect to $\tilde{\omega}$

$$\tilde{\mathbf{f}}_{a/\tilde{\omega}} = \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c \left(\tilde{\mathbf{c}}_{f_{a/y}} y_{/\tilde{\omega}} \right) \quad (12.171)$$

where:

$$y_{/\tilde{\omega}} = \mathbf{D} u_{/\tilde{\omega}} \quad (12.172)$$

Perturbation of $\tilde{\mathbf{f}}_a$ with respect to \mathbf{q}

$$\tilde{\mathbf{f}}_{a/\mathbf{q}} = \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c \left(\tilde{\mathbf{c}}_{f_{a/y}} \mathbf{y}_{/\mathbf{q}} \right) \quad (12.173)$$

where:

$$\mathbf{y}_{/\mathbf{q}} = \mathbf{C} \quad (12.174)$$

Perturbation of \tilde{f}_a with respect to \dot{q}

$$\tilde{f}_{a/\dot{q}} = \mathbf{0} \quad (12.175)$$

12.7.7 Perturbation of the aerodynamic moments

Perturbation of \tilde{c}_a with respect to \tilde{v}

$$\tilde{c}_{a/\tilde{v}} = \rho c^2 \tilde{\mathbf{c}}_{c_a} \begin{bmatrix} \tilde{v}_x & \tilde{v}_y & \tilde{v}_z \end{bmatrix} + \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c^2 \left(\tilde{\mathbf{c}}_{c_a/y} y/\tilde{v} + \tilde{\mathbf{c}}_{c_a/U_\infty} U_\infty/\tilde{v} \right) \quad (12.176)$$

$$\tilde{\mathbf{c}}_{c_a/U_\infty} = \begin{Bmatrix} 0 \\ 0 \\ \frac{c_{l_\alpha}}{2} \frac{c}{2} \frac{1}{4U_\infty^2} \left(\dot{\alpha}_{pivot} - \left(\frac{ca}{U_\infty} - \frac{c}{4U_\infty} \right) \ddot{\alpha} + \dot{\alpha} \right) \end{Bmatrix} \quad (12.177)$$

$$\tilde{\mathbf{c}}_{c_a/y} = \begin{bmatrix} 0 \\ 0 \\ \frac{dc_m^{\text{lookup}}}{d\alpha} \end{bmatrix} \quad (12.178)$$

where again the dependence of $\dot{\alpha}_{pivot}$, $\dot{\alpha}$ and $\ddot{\alpha}$ is neglected.

Perturbation of \tilde{c}_a with respect to $\tilde{\omega}$

$$\tilde{c}_{a/\tilde{\omega}} = \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c^2 \left(\tilde{\mathbf{c}}_{c_a/y} y/\tilde{\omega} \right) \quad (12.179)$$

Perturbation of \tilde{c}_a with respect to q

$$\tilde{c}_{a/q} = \frac{1}{2} \rho \tilde{\mathbf{v}}^T \tilde{\mathbf{v}} c^2 \left(\tilde{\mathbf{c}}_{c_a/y} y/q \right) \quad (12.180)$$

Perturbation of \tilde{c}_a with respect to \dot{q}

$$\tilde{c}_{a/\dot{q}} = \mathbf{0} \quad (12.181)$$

12.8 Rotor

Definitions. Axis 3 is the rotor's axis. $\tilde{\mathbf{v}}$ is the composition of the velocity of the 'aircraft' node and of the airstream speed, if any, projected in the reference frame of the 'aircraft' node, namely

$$\tilde{\mathbf{v}} = \mathbf{R}_{\text{craft}}^T (-\mathbf{v}_{\text{craft}} + \mathbf{v}_\infty). \quad (12.182)$$

$$v_{12} = \sqrt{v_1^2 + v_2^2} \quad (12.183a)$$

$$v = \sqrt{v_1^2 + v_2^2 + v_3^2} = \sqrt{\tilde{\mathbf{v}}^T \tilde{\mathbf{v}}} \quad (12.183b)$$

$$\sin \alpha_d = -v_3/v \quad (12.183c)$$

$$\cos \alpha_d = v_{12}/v \quad (12.183d)$$

$$\psi_0 = \tan^{-1} \left(\frac{v_2}{v_1} \right) \quad (12.183e)$$

$$v_{\text{tip}} = \Omega R \quad (12.183f)$$

$$\mu = \cos \alpha_d \frac{v}{v_{\text{tip}}} \quad (12.183g)$$

Note: $v \geq 0$ and $v_{12} \geq 0$ by definition; as a consequence, $\cos \alpha_d \geq 0$, while the sign of $\sin \alpha_d$ depends on whether the flow related to the absolute motion of the rotor enters the disk from above (> 0) or from below (< 0). $v_{\text{tip}} > 0$ by construction ($\Omega = \|\boldsymbol{\omega}\|$, and no induced velocity is computed if Ω is below a threshold). As a consequence, $\mu \geq 0$.

Ground effect. If defined, according to [7],

$$u_{\text{IGE}} = k_{\text{GE}} u_{\text{OGE}} \quad (12.184a)$$

$$k = 1 - \frac{1}{z^2} \quad (12.184b)$$

$$z = \max \left(\frac{h}{R}, \frac{1}{4} \right) \quad (12.184c)$$

where h is the component along axis 3 of the ‘ground’ node of distance between the ‘aircraft’ and the ‘ground’ node, assuming the ‘aircraft’ node is located at the hub center.

Reference Induced Velocity. The reference induced velocity u is computed by solving the implicit problem

$$f = \lambda_u - \frac{C_t}{2\sqrt{\mu^2 + \lambda^2}} = 0, \quad (12.185)$$

with

$$C_t = \frac{T}{\rho \pi R^4 \Omega^2} \quad (12.186a)$$

$$\lambda_u = \frac{u}{v_{\text{tip}}} \quad (12.186b)$$

$$\lambda = \frac{v \sin \alpha_d + u}{v_{\text{tip}}} = \mu \tan \alpha_d + \lambda_u; \quad (12.186c)$$

T is the component of the aerodynamic force of the rotor along the shaft axis. The value of λ_u is initialized using the reference induced velocity u at the previous step/iteration. Only when $u = 0$ and $C_t \neq 0$, u is initialized using its nominal value in hover,

$$u = \text{sign}(T) \sqrt{\frac{\|T\|}{2\rho A}}. \quad (12.187)$$

The problem is solved by means of a local Newton iteration. The Jacobian of the problem is

$$\frac{\partial f}{\partial \lambda_u} = 1 + \frac{C_t}{2(\mu^2 + \lambda^2)^{3/2}} \lambda. \quad (12.188)$$

The solution,

$$\Delta \lambda_u = - \left(\frac{\partial f}{\partial \lambda_u} \right)^{-1} f, \quad (12.189)$$

is added to λ_u as $\lambda_u+ = \eta \Delta \lambda_u$, where $0 < \eta \leq 1$ is an optional relaxation factor (by default, $\eta = 1$).

Corrections. The reference induced velocity is corrected by separately correcting the inflow and advance parameters, namely

$$\lambda^* = \frac{\lambda}{k_H^2} \quad (12.190a)$$

$$\mu^* = \frac{\mu}{k_{FF}} \quad (12.190b)$$

(by default, $k_H = 1$ and $k_{FF} = 1$). The reference induced velocity is then recomputed as

$$u^* = (1 - \rho) k_{GE} v_{tip} \frac{C_t}{2\sqrt{\mu^{*2} + \lambda^{*2}}} + \rho u_{prev}^*, \quad (12.191)$$

where $0 \leq \rho < 1$ is a memory factor (by default, $\rho = 0$).

Note: in principle, multiple solutions for λ_u are possible. However, only one solution is physical. Currently, no strategy is put in place to ensure that only the physical solution is considered.

12.8.1 Uniform Inflow Model

The induced velocity is equal to its reference value, u^* , everywhere.

12.8.2 Glauert Model

In forward flight (when $\mu > 0.15$) the inflow over the rotor disk can be approximated by:

$$\lambda_i = \lambda_0 \left(1 + k_x \frac{x}{R} + k_y \frac{y}{R} \right) \quad (12.192)$$

$$= \lambda_0 (1 + k_x r \cos \psi + k_y r \sin \psi), \quad (12.193)$$

where the mean induced velocity λ_0 is computed as shown in the previous section, while $r = \sqrt{x^2 + y^2}/R$ is the nondimensional radius.

In literature a lot of expressions for the k_x and k_y coefficients have been proposed by different authors, as summarized in table 12.2. Up to now in MBDyn the following expressions are implemented:

$$k_x = \frac{4}{3} (1 - 1.8\mu^2) \tan \frac{\chi}{2} \quad (12.194)$$

$$k_y = 0, \quad (12.195)$$

FIXME: è diversa dalle espressioni riportate sul Leishman!

Table 12.2: Glauert inflow model (source: Leishman [2])

Author(s)	k_x	k_y
Coleman et al. (1945)	$\tan \frac{\chi}{2}$	0
Drees (1949)	$\frac{4}{3} \frac{(1 - \cos \chi - 1.8\mu^2)}{\sin \chi}$	-2μ
Payne (1959)	$\frac{4}{3} (\mu/\lambda)/(1.2 + \mu/\lambda)$	0
White & Blake (1979)	$\sqrt{2} \sin \chi$	0
Pitt & Peters (1981)	$\frac{15\pi}{23} \tan \frac{\chi}{2}$	0
Howlett (1981)	$\sin^2 \chi$	0

where χ is the wake skew angle:

$$\chi = \tan^{-1} \left(\frac{\mu}{\lambda} \right). \quad (12.196)$$

TODO: check, dovrebbe essere equivalente all'espressione di Leishman ma sarebbe meglio verificare!!!

Note: the Glauert inflow model exactly matches the uniform inflow model when the advance ratio is null (in hover). In fact, when $\mu = 0$ then $\chi = k_x = 0$. The inflow is thus uniform over the rotor disk, and equal to λ_0 .

Table 12.3: Glauert inflow model as implemented in MBDyn

Name	Author(s)	k_x	k_y
(default)	Glauert	$\frac{4}{3} (1 - 1.8\mu^2) \tan \left(\frac{\chi}{2} \right)$	0
coleman	Coleman et al. (1945)	$\tan \left(\frac{\chi}{2} \right)$	0
drees	Drees (1949)	$\frac{4}{3} \frac{(1 - \cos \chi - 1.8\mu^2)}{\sin \chi}$	-2μ
payne	Payne (1959)	$\frac{4}{3} \frac{\mu/\lambda}{1.2 + \mu/\lambda}$	0
white and blake	White & Blake (1979)	$\sqrt{2} \sin \chi$	0
pitt and peters	Pitt & Peters (1981)	$\frac{15\pi}{23} \tan \left(\frac{\chi}{2} \right)$	0
howlett	Howlett (1981)	$\sin^2 \chi$	0
drees 2	Drees (?)	$\frac{4}{3} (1 - 1.8\mu^2) \sqrt{1 - \frac{\lambda}{\mu} + \left(\frac{\lambda}{\mu} \right)^2}$	-2μ

12.8.3 Mangler-Squire Model

The Mangler-Squire model is developed under the high speed assumption and it should be used only for advance ratio grater than 0.1.

In the original formulation the induced velocity is computed as:

$$\lambda_i = \left(\frac{2C_T}{\mu} \right) \left[\frac{c_0}{2} - \sum_{n=1}^{\infty} c_n(r, \alpha_d) \cos n\psi \right], \quad (12.197)$$

since the advance ratio μ appears in the denominator this expression is not valid in hover. Bramwell [8] proposed a different expression for the induced velocity:

$$\lambda_i = 4\lambda_0 \left[\frac{c_0}{2} - \sum_{n=1}^{\infty} c_n(r, \alpha_d) \cos n\psi \right], \quad (12.198)$$

where λ_0 is the mean inflow computed as shown before. In this way the Mangler-Squire inflow model makes sense also in hover. MBDyn uses the latter version.

The expression of the c_n coefficients depends on the form of the rotor disk loading. Mangler and Squire developed the theory for two fundamental forms: Type I (elliptical loading) and Type III (a loading that vanishes at the edges and at the center of the disk). The total loading is finally obtained by a linear combination of Type I and Type III loadings (see [2]).

MBDyn uses just a Type III loading, and the resulting expressions for the c_n coefficients are:

$$c_0 = \frac{15}{8}\eta(1-\eta^2), \quad (12.199)$$

where $\eta = \sqrt{1-r^2}$, and $r = \sqrt{x^2+y^2}/R$ is the nondimensional radius.

$$c_1 = -\frac{15\pi}{256}(5-9\eta^2)\left[(1-\eta^2)\left(\frac{1-\sin\alpha_d}{1+\sin\alpha_d}\right)\right]^{\frac{1}{2}}, \quad (12.200)$$

$$c_3 = \frac{45\pi}{256}\left[(1-\eta^2)\left(\frac{1-\sin\alpha_d}{1+\sin\alpha_d}\right)\right]^{\frac{3}{2}}, \quad (12.201)$$

and $c_n = 0$ for odd values of $n \geq 0$.

For even values:

$$c_n = (-1)^{\frac{n}{2}-1}\frac{15}{8}\left[\frac{\eta+n}{n^2-1}\frac{9\eta^2+n^2-6}{n^2-9} + \frac{3\eta}{n^2-9}\right]\left[\left(\frac{1-\eta}{1+\eta}\right)\left(\frac{1-\sin\alpha_d}{1+\sin\alpha_d}\right)\right]^{\frac{n}{2}}, \quad (12.202)$$

Note: the version proposed by Bramwell [8] makes sense also in hover but gives different results with respect to the uniform inflow and the Glauert inflow models. Let assume $\alpha_d = \pi/2$, it follows that $c_n = 0$ for $n \geq 1$. Therefore the induced velocity does not depend on the azimuthal position ψ but only on the radial position r , so the inflow is not uniform, but the mean inflow is still λ_0 .

12.8.4 Dynamic Inflow Model

The dynamic inflow model implemented in MBDyn has been developed by Pitt and Peters [9]. Here the model is just briefly described together with the MBDyn implementation peculiarities.

The inflow dynamics is represented by a simple first-order linear model:

$$M\dot{\lambda} + L^{-1}\lambda = c, \quad (12.203)$$

where λ is a vector with 3 elements:

$$\lambda = \begin{bmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{bmatrix}, \quad (12.204)$$

the induced velocity on the rotor disk is finally obtained as function of the azimuthal angle ψ and the non dimensional radial position $r = \frac{\sqrt{x^2+y^2}}{R}$ using the following equation:

$$u_{ind}(r, \psi) = \Omega R(\lambda_0 + \lambda_s r \sin \psi + \lambda_c r \cos \psi). \quad (12.205)$$

The right-hand term in equation 12.203 contains the thrust, roll moment and pitch moment coefficients:

$$\mathbf{c} = \begin{bmatrix} C_T \\ C_L \\ C_M \end{bmatrix} = \begin{bmatrix} \frac{T}{\rho A \Omega^2 R^2} \\ \frac{L}{\rho A \Omega^2 R^3} \\ \frac{M}{\rho A \Omega^2 R^3} \end{bmatrix}, \quad (12.206)$$

while $\boldsymbol{\lambda}$ is derived with respect a non-dimensional time Ωt :

$$\dot{\boldsymbol{\lambda}} = \frac{d\boldsymbol{\lambda}}{d(\Omega t)}. \quad (12.207)$$

Equation 12.203 could be rewritten as:

$$\mathbf{M} \dot{\boldsymbol{\lambda}} + \Omega \mathbf{L}^{-1} \boldsymbol{\lambda} = \Omega \mathbf{c}, \quad (12.208)$$

where now the dot represents the (dimensional) time derivative. The latter is the form implemented in MBDyn.

Matrix \mathbf{L} is defined as:

$$\mathbf{L} = \tilde{\mathbf{L}} \cdot \mathbf{K} = \begin{bmatrix} \frac{1}{2} & 0 & \frac{15\pi}{64} \tan \frac{\chi}{2} \\ 0 & -\frac{4}{2 \cos^2 \frac{\chi}{2}} & 0 \\ \frac{15\pi}{64} \tan \frac{\chi}{2} & 0 & -\frac{4(2 \cos^2 \frac{\chi}{2} - 1)}{2 \cos^2 \frac{\chi}{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{V_T} \\ \frac{1}{V_m} \\ \frac{1}{V_m} \end{bmatrix}, \quad (12.209)$$

where χ is wake skew angle defined as:

$$\chi = \tan^{-1} \frac{\mu}{\lambda}, \quad (12.210)$$

where

$$\mu = \frac{V_\infty \cos \alpha_d}{\Omega R}, \quad (12.211)$$

and

$$\lambda = \frac{V_\infty \sin \alpha_d}{\Omega R} + \frac{u_{ind}^0}{\Omega R}, \quad (12.212)$$

where u_{ind}^0 is the steady uniform induced velocity computed as described in the uniform inflow model section. The elements in matrix \mathbf{K} matrix are respectively:

$$V_T = \sqrt{\lambda^2 + \mu^2}, \quad (12.213)$$

and

$$V_m = \frac{\mu^2 + \lambda \left(\lambda + \frac{u_{ind}^0}{\Omega R} \right)}{\sqrt{\lambda^2 + \mu^2}} \quad (12.214)$$

Note: matrix L is slightly different from matrix L in Pitt and Peters' paper [9] because here the first column elements are divided by V_T while the second and third columns elements by V_m , whereas in matrix L an unique value v is used for all the elements in the matrix. The V_m term corresponds to the *steady lift mass-flow parameter* defined in the Pitt and Peters paper, while the V_T corresponds to the *no lift mass-flow parameter*, because $\bar{\lambda} + \bar{\nu} = \lambda$ and $\bar{\nu} = u_{ind}^0 / (\Omega R)$. Moreover in the Pitt and Peters work the elements are function of $\alpha = \tan^{-1} \frac{\lambda}{\mu}$, the relation between α and χ is:

$$\alpha = \frac{\pi}{2} - \chi. \quad (12.215)$$

So,

$$\sin \alpha = \cos \chi = \cos \left(\frac{\chi}{2} + \frac{\chi}{2} \right) = \cos^2 \frac{\chi}{2} - \sin^2 \frac{\chi}{2} = 2 \cos^2 \frac{\chi}{2} - 1, \quad (12.216)$$

$$\sqrt{\frac{1 - \sin \alpha}{1 + \sin \alpha}} = \sqrt{\frac{2 - 2 \cos^2 \frac{\chi}{2}}{2 \cos^2 \frac{\chi}{2}}} = \sqrt{\frac{2 \sin^2 \frac{\chi}{2}}{2 \cos^2 \frac{\chi}{2}}} = \tan \frac{\chi}{2}. \quad (12.217)$$

That means that the only difference in the MBDyn implementation is related to matrix K .

In MBDyn the inversion of the L matrix is formulated analytically; matrix L^* is defined as:

$$L^* = \Omega L^{-1} = \Omega \begin{bmatrix} \frac{l_{33}}{l_{11}l_{33} - l_{13}l_{31}} & 0 & -\frac{l_{13}}{l_{11}l_{33} - l_{13}l_{31}} \\ 0 & \frac{1}{l_{22}} & 0 \\ -\frac{l_{31}}{l_{11}l_{33} - l_{13}l_{31}} & 0 & \frac{l_{11}}{l_{11}l_{33} - l_{13}l_{31}} \end{bmatrix}. \quad (12.218)$$

Finally, matrix M is defined as:

$$M = \begin{bmatrix} \frac{128}{75\pi} & 0 & 0 \\ 0 & -\frac{16}{45\pi} & 0 \\ 0 & 0 & -\frac{16}{45\pi} \end{bmatrix}. \quad (12.219)$$

Note: this choice of matrix M corresponds to the mixed *uncorrected-corrected* (following the authors' nomenclature) M-matrix proposed by Pitt and Peters in their paper.

12.9 Aero Modal

12.9.1 Clamped

TODO

The matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D}_0 , \mathbf{D}_1 and \mathbf{D}_2 , of a state space model according to the representation

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bq} \\ \mathbf{f} &= q \left(\mathbf{Cx} + \mathbf{D}_0\mathbf{q} + \frac{c}{2V_\infty} \mathbf{D}_1\dot{\mathbf{q}} + \left(\frac{c}{2V_\infty} \right)^2 \mathbf{D}_2\ddot{\mathbf{q}} \right)\end{aligned}$$

where \mathbf{x} are the aerodynamic state variables, \mathbf{q} are the modal variables that describe the structural motion as defined in the related modal joint, c is the reference length, V_∞ is the free airstream velocity, $q = \rho V_\infty^2 / 2$ is the dynamic pressure, and \mathbf{f} are the unsteady aerodynamic forces applied to the structural dynamics equations.

12.9.2 Free

TODO

The matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D}_0 , \mathbf{D}_1 and \mathbf{D}_2 , of a state space model according to the representation

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{B}_x\mathbf{q}_x + \mathbf{B}_\theta\mathbf{q}_\theta \\ \mathbf{f} &= q \left(\mathbf{Cx} + \mathbf{D}_0\mathbf{q} + \mathbf{D}_{0q_x}\mathbf{q}_x + \mathbf{D}_{0q_\theta}\mathbf{q}_\theta + \frac{c}{2V_\infty} (\mathbf{D}_1\dot{\mathbf{q}} + \mathbf{D}_{1q_x}\dot{\mathbf{q}}_x + \mathbf{D}_{1q_\theta}\dot{\mathbf{q}}_\theta) + \left(\frac{c}{2V_\infty} \right)^2 (\mathbf{D}_2\ddot{\mathbf{q}} + \mathbf{D}_{2q_x}\ddot{\mathbf{q}}_x + \mathbf{D}_{2q_\theta}\ddot{\mathbf{q}}_\theta) \right) \\ \mathbf{F} &= q\mathbf{R} \left(\mathbf{C}_F\mathbf{x} + \mathbf{D}_{0F}\mathbf{q} + \mathbf{D}_{0Fq_x}\mathbf{q}_x + \mathbf{D}_{0Fq_\theta}\mathbf{q}_\theta + \frac{c}{2V_\infty} (\mathbf{D}_{1F}\dot{\mathbf{q}} + \mathbf{D}_{1Fq_x}\dot{\mathbf{q}}_x + \mathbf{D}_{1Fq_\theta}\dot{\mathbf{q}}_\theta) + \left(\frac{c}{2V_\infty} \right)^2 (\mathbf{D}_{2F}\ddot{\mathbf{q}} + \mathbf{D}_{2Fq_x}\ddot{\mathbf{q}}_x + \mathbf{D}_{2Fq_\theta}\ddot{\mathbf{q}}_\theta) \right) \\ \mathbf{M} &= \end{aligned}$$

where $\mathbf{q}_x = \mathbf{R}^T(\mathbf{x} - \mathbf{x}_0)$, $\mathbf{q}_\theta = \mathbf{R}^T(\boldsymbol{\theta}_\Delta - \boldsymbol{\theta}_{\Delta 0})$, $\dot{\mathbf{q}}_x = \mathbf{R}^T\dot{\mathbf{x}}$, $\dot{\mathbf{q}}_\theta = \mathbf{R}^T\boldsymbol{\omega}$, ...

Chapter 13

Forces

13.1 Abstract Force

13.1.1 Abstract

13.1.2 Abstract Internal

13.2 Structural Forces

13.2.1 Force

Absolute

Follower

Absolute Internal

Follower Internal

13.2.2 Couple

Absolute

Follower

Absolute Internal

Follower Internal

13.3 Modal

13.4 External

The external structural force element allows MBDyn to cooperate with external solvers by defining a meta-element that applies forces and moments to a set of structural nodes. The forces and moments are provided by an external solver, called the *peer*. MBDyn provides the peer information about the motion of the nodes participating in the set.

Optionally, an additional layer of field mapping can be interposed. In this case, the motion of the nodes is transformed in the motion of a set of intermediate points, which is further mapped into the motion of the points known by the peer by means of a linear mapping.

The forces at the mapped points returned by the peer are mapped back into forces and moments for the structural nodes participating in the set of the element.

13.4.1 External Structural

The external structural and external structural mapping forces can be formulated directly in the absolute frame, or referred to a reference node. In the former case, operations are straightforward; in the latter one, the kinematics are first expressed in the reference frame of the reference node and then sent to the peer along with the motion of the reference node. The latter returns nodal forces and moments oriented according to the reference frame of the reference node. The additional operations performed by the mapping variant are discussed separately in a subsequent section. The orientation of the reference node is \mathbf{R}_r ; the position is \mathbf{x}_r .

The orientation of the generic node i is

$$\mathbf{R}_i = \mathbf{R}_r \bar{\mathbf{R}}_i. \quad (13.1)$$

The relative orientation passed to the peer solver is

$$\bar{\mathbf{R}}_i = \mathbf{R}_r^T \mathbf{R}_i. \quad (13.2)$$

The position of the generic node i is

$$\mathbf{x}_i = \mathbf{x}_r + \mathbf{R}_r \bar{\mathbf{x}}_i. \quad (13.3)$$

The relative position passed to the peer solver is

$$\bar{\mathbf{x}}_i = \mathbf{R}_r^T (\mathbf{x}_i - \mathbf{x}_r). \quad (13.4)$$

The angular velocity of the generic node i is

$$\boldsymbol{\omega}_i \times = \dot{\mathbf{R}}_i \mathbf{R}_i^T = \dot{\mathbf{R}}_r \mathbf{R}_r^T + \mathbf{R}_r \dot{\bar{\mathbf{R}}}_i \bar{\mathbf{R}}_i^T \mathbf{R}_r^T = \boldsymbol{\omega}_r \times + \mathbf{R}_r \bar{\boldsymbol{\omega}}_i \times \mathbf{R}_r^T. \quad (13.5)$$

The relative angular velocity passed to the peer solver is

$$\bar{\boldsymbol{\omega}}_i = \mathbf{R}_r^T (\boldsymbol{\omega}_i - \boldsymbol{\omega}_r). \quad (13.6)$$

The velocity of the generic node i is

$$\dot{\mathbf{x}}_i = \dot{\mathbf{x}}_r + \boldsymbol{\omega}_r \times \mathbf{R}_r \bar{\mathbf{x}}_i + \mathbf{R}_r \dot{\bar{\mathbf{x}}}_i. \quad (13.7)$$

The relative velocity passed to the peer solver is

$$\dot{\bar{\mathbf{x}}}_i = \mathbf{R}_r^T (\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_r) - (\mathbf{R}_r^T \boldsymbol{\omega}_r) \times \bar{\mathbf{x}}_i \quad (13.8)$$

TODO: accelerations

The virtual work done by the forces applied to the nodes is

$$\delta \mathcal{L} = \delta \mathbf{x}_r \cdot \mathbf{f}_r + \boldsymbol{\theta}_{r\delta} \cdot \mathbf{m}_r + \sum_i (\delta \bar{\mathbf{x}}_i \cdot \bar{\mathbf{f}}_i + \bar{\boldsymbol{\theta}}_{i\delta} \cdot \bar{\mathbf{m}}_i). \quad (13.9)$$

The virtual rotation of the generic node i is

$$\delta \mathbf{R}_i \mathbf{R}_i^T = \boldsymbol{\theta}_{i\delta} \times = \delta \mathbf{R}_r \mathbf{R}_r^T + \mathbf{R}_r \delta \bar{\mathbf{R}}_i \bar{\mathbf{R}}_i^T \mathbf{R}_r^T = \boldsymbol{\theta}_{r\delta} \times + \mathbf{R}_r \bar{\boldsymbol{\theta}}_{i\delta} \times \mathbf{R}_r^T. \quad (13.10)$$

The relative virtual rotation is thus

$$\bar{\boldsymbol{\theta}}_{i\delta} = \mathbf{R}_r^T (\boldsymbol{\theta}_{i\delta} - \boldsymbol{\theta}_{r\delta}). \quad (13.11)$$

The virtual displacement of the generic node i is

$$\delta \mathbf{x}_i = \delta \mathbf{x}_r + \boldsymbol{\theta}_{r\delta} \times \mathbf{R}_r \bar{\mathbf{x}}_i + \mathbf{R}_r \delta \bar{\mathbf{x}}_i. \quad (13.12)$$

The relative virtual displacement is thus

$$\delta \bar{\mathbf{x}}_i = \mathbf{R}_r^T (\delta \mathbf{x}_i - \delta \mathbf{x}_r) + \bar{\mathbf{x}}_i \times (\mathbf{R}_r^T \boldsymbol{\theta}_{r\delta}). \quad (13.13)$$

The virtual work becomes

$$\delta \mathcal{L} = \delta \mathbf{x}_r \cdot \mathbf{f}_r + \boldsymbol{\theta}_{r\delta} \cdot \mathbf{m}_r + \sum_i (\delta \mathbf{x}_i \cdot \mathbf{R}_r \bar{\mathbf{f}}_i + \boldsymbol{\theta}_{i\delta} \cdot \mathbf{R}_r \bar{\mathbf{m}}_i - \delta \mathbf{x}_r \cdot \mathbf{R}_r \bar{\mathbf{f}}_i - \boldsymbol{\theta}_{r\delta} \cdot \mathbf{R}_r (\bar{\mathbf{m}}_i + \bar{\mathbf{x}}_i \times \bar{\mathbf{f}}_i)). \quad (13.14)$$

The force and moment acting on the generic node i are

$$\mathbf{f}_i = \mathbf{R}_r \bar{\mathbf{f}}_i \quad (13.15)$$

$$\mathbf{m}_i = \mathbf{R}_r \bar{\mathbf{m}}_i. \quad (13.16)$$

The moment is always intrinsically referred to the current position of the node. The force and moment acting on the reference node are

$$\mathbf{f} = \mathbf{f}_r - \sum_i \mathbf{R}_r \bar{\mathbf{f}}_i \quad (13.17)$$

$$\mathbf{m} = \mathbf{m}_r - \sum_i \mathbf{R}_r (\bar{\mathbf{m}}_i + \bar{\mathbf{x}}_i \times \bar{\mathbf{f}}_i). \quad (13.18)$$

The moment is always intrinsically referred to the current position of the node.

In principle, \mathbf{f} and \mathbf{m} should be identically zero, unless the reference node is specifically loaded. This fact can be exploited by setting `use reference node forces to no`, which results in ignoring \mathbf{f} and \mathbf{m} .

13.4.2 External Structural Mapping

The external structural mapping case differs from the previous one in the fact that two additional intermediate layers are added. The first layer computes the motion of a set of points rigidly connected to the nodes that participate in the set. Each node n can have an arbitrary number of associated points p , whose position in the reference frame of the node is $\tilde{\mathbf{o}}_{np}$. The position of point p associated to node n is

$$\mathbf{x}_{np} = \mathbf{x}_n + \mathbf{R}_n \tilde{\mathbf{o}}_{np}. \quad (13.19)$$

The position \mathbf{x}_{np} , instead of \mathbf{x}_i , must be used to compute the relative position. Orientations are not mapped.

As soon as the positions, velocities (and accelerations, if needed) of the points are computed and collected in vectors denoted by the subscript $(\cdot)_{\text{mbdyn}}$, they are mapped into the corresponding quantities of the peer, denoted by the subscript $(\cdot)_{\text{peer}}$, using a linear transformation \mathbf{H} , namely

$$\mathbf{x}_{\text{peer}} = \mathbf{H}\mathbf{x}_{\text{mbdyn}} \quad (13.20)$$

$$\dot{\mathbf{x}}_{\text{peer}} = \mathbf{H}\dot{\mathbf{x}}_{\text{mbdyn}} \quad (13.21)$$

$$\delta\mathbf{x}_{\text{peer}} = \mathbf{H}\delta\mathbf{x}_{\text{mbdyn}}. \quad (13.22)$$

The work done in the peer domain must be equal by the work done in MBDyn's; this implies

$$\delta\mathcal{L}_{\text{mbdyn}} = \delta\mathbf{x}_{\text{mbdyn}} \cdot \mathbf{f}_{\text{mbdyn}} = \delta\mathbf{x}_{\text{peer}} \cdot \mathbf{f}_{\text{peer}} = \delta\mathcal{L}_{\text{peer}}, \quad (13.23)$$

which yields

$$\delta\mathbf{x}_{\text{mbdyn}} \cdot \mathbf{f}_{\text{mbdyn}} = \delta\mathbf{x}_{\text{mbdyn}} \cdot \mathbf{H}^T \mathbf{f}_{\text{peer}}, \quad (13.24)$$

and thus, thanks to the arbitrariness of virtual displacements,

$$\mathbf{f}_{\text{mbdyn}} = \mathbf{H}^T \mathbf{f}_{\text{peer}}. \quad (13.25)$$

The forces $\mathbf{f}_{\text{mbdyn}}$ correspond to the points of the intermediate mapping layer. They are transformed into the corresponding nodal forces and moments considering the work done by the virtual perturbation of Eq. (13.19),

$$\delta\mathbf{x}_{np} = \delta\mathbf{x}_n - (\mathbf{R}_n \tilde{\mathbf{o}}_{np}) \times \boldsymbol{\theta}_{n\delta}, \quad (13.26)$$

namely

$$\delta\mathcal{L}_n = \sum_p \delta\mathbf{x}_{np} \cdot \mathbf{f}_{np} = \sum_p (\delta\mathbf{x}_n \cdot \mathbf{f}_{np} + \boldsymbol{\theta}_{n\delta} \cdot (\mathbf{R}_n \tilde{\mathbf{o}}_{np}) \times \mathbf{f}_{np}). \quad (13.27)$$

The corresponding nodal force and moment are

$$\mathbf{f}_n = \sum_p \mathbf{f}_{np} \quad (13.28)$$

$$\mathbf{m}_n = \sum_p (\mathbf{R}_n \tilde{\mathbf{o}}_{np}) \times \mathbf{f}_{np}. \quad (13.29)$$

The moment is always intrinsically referred to the current position of the node.

When a reference node is defined, all symbols in the expressions of the nodal force and moment must bear an overbar ($\bar{\cdot}$), to indicate that they are relative and thus need to be further transformed as previously shown for the external structural force element.

13.4.3 External Modal

13.4.4 External Modal Mapping

13.4.5 Client Library

The peer side of the communication protocol has been implemented in a set of client libraries. At the core there is `libmbc`, a library written in C that performs the core operations. Declarations are provided in the header file `mbc.h`.

A high-level interface is provided in C++ in `libmbcxx`. Declarations are provided in the header file `mbcxx.h`.

Another interface is provided in Python, in module `_mbc_py.so`. The corresponding Python code is defined in `mbc_py_interface.py`.

Socket-Based Protocol

Each communication is prefixed by a `uint8_t` value that indicates the type of operation being performed. It is optionally followed by a message. Legal values are

- `ES_REGULAR_DATA`
- `ES_GOTO_NEXT_STEP`
- `ES_ABORT`
- `ES_REGULAR_DATA_AND_GOTO_NEXT_STEP`
- `ES_NEGOTIATION`
- `ES_OK`

The corresponding numerical value is defined in `mbc.h`.

Operations are:

- negotiation: the client tells the server what
- kinematics exchange: the master sends the motion
- forces exchange: the peer receives the loads

Chapter 14

Hydraulic Library

14.1 Hydraulic Fluids

14.2 Hydraulic Nodes

14.3 Hydraulic Elements

14.3.1 Accumulator

The accumulator defines two internal states x and v that represent the position and the velocity of the cap that, in a conventional gas device, separates the fluid and the gas. However, both the gravity effect and a linear spring effect can be considered as well, and any combination of reaction forces can be modeled by setting the appropriate parameters: g for a gravity device, p_{g0} for a gas device, and k for a linear spring device.

$$\begin{aligned} 0 &= q \\ m\dot{v} + kx &= -mg + Ap(p - p_g) - f_0 - \frac{1}{2}\rho A c_e \left(\frac{A}{A_p}\right)^2 |v| v \\ &\quad - \text{step}(x_{\min} - x)(c_1(x - x_{\min}) + c_2v + c_3\dot{v}) \\ &\quad - \text{step}(x - x_{\max})(c_1(x - x_{\max}) + c_2v + c_3\dot{v}) \\ \dot{x} &= v \end{aligned}$$

where $p_g = p_{g0} \left(\frac{l}{l-x}\right)^\gamma$ and $q = \rho Av$.

14.3.2 Actuator

The hydraulic actuator element couples the hydraulic library with the structural library. It connects the displacement of two structural nodes to the flow through two hydraulic nodes, and the pressure at two hydraulic nodes to the forces applied at two structural nodes. In the spirit of the multibody analysis philosophy, this element provides the essential connection between structural and hydraulic nodes; the constraints between the structural nodes, and other flow elements, e.g. leakages between the chambers, must be added by the user.

Definitions

In the following, $(\cdot)_{s1}$ and $(\cdot)_{s2}$ refer to structural nodes 1 and 2, and $(\cdot)_{h1}$ and $(\cdot)_{h2}$ refer to hydraulic nodes 1 and 2. The structural node labeled as 1 is assumed as the cylinder, and its orientation determines the axis of the actuator. The relative orientation of the actuator is defined by the unit vector $\tilde{\mathbf{v}}$, and the absolute orientation is $\mathbf{R}_{s1}\tilde{\mathbf{v}}$. It is assumed that appropriate kinematic constraints allow only a relative displacement of the structural nodes along the axis $\tilde{\mathbf{v}}$, and the only relative rotation, if any, is about the axis itself. This can be obtained by combining an inline joint with a revolute rotation or a prismatic joint.

Equations

$$\begin{aligned} 0 &= -\mathbf{F} \\ 0 &= -\mathbf{f}_{s1} \times \mathbf{F} \\ 0 &= \mathbf{F} \\ 0 &= \mathbf{f}_{s2} \times \mathbf{F} \\ 0 &= q_{h1} \\ 0 &= q_{h2} \\ p_{h1} &= P_{h1} \\ p_{h2} &= P_{h2} \end{aligned}$$

The first four equations apply the force resulting from the hydraulic pressure to the structural nodes; the fifth and the sixth apply the flow resulting from the actuator kinematics to the flow balance equations of the hydraulic nodes. The last two equations are required to associate two scalar differential unknowns to the hydraulic node pressures, because the flow definitions require the derivative of the pressure, while the hydraulic nodes are defined as scalar algebraic.

The force is defined as

$$\mathbf{F} = (A_{h1}p_{h1} - A_{h2}p_{h2})(\mathbf{R}_{s1}\tilde{\mathbf{v}}) \quad (14.1)$$

The distance between the structural nodes, along the actuator axis, is

$$l = (\mathbf{R}_{s1}\tilde{\mathbf{v}})^T(\mathbf{x}_{s2} + \mathbf{f}_{s2} - \mathbf{x}_{s1} - \mathbf{f}_{s1}) \quad (14.2)$$

The relative velocity of the structural nodes, along the actuator axis, is

$$\dot{l} = (\mathbf{R}_{s1}\tilde{\mathbf{v}})^T(\dot{\mathbf{x}}_{s2} - \dot{\mathbf{x}}_{s1} + \boldsymbol{\omega}_{s1} \times (\mathbf{x}_{s2} - \mathbf{x}_{s1}) + (\boldsymbol{\omega}_{s2} - \boldsymbol{\omega}_{s1}) \times \mathbf{f}_{s2}) \quad (14.3)$$

The flow at the two hydraulic nodes is

$$\begin{aligned} q_{h1} &= A_{h1} \left(l \frac{\partial \rho_{h1}}{\partial p} \dot{P}_{h1} + \rho_{h1} \dot{d} \right) \\ q_{h2} &= A_{h2} \left((L - l) \frac{\partial \rho_{h2}}{\partial p} \dot{P}_{h2} - \rho_{h2} \dot{d} \right) \end{aligned}$$

where ρ is the fluid density (different fluids in the chambers are allowed), and L is the total length of the actuator. In case stroke limitations must be enforced, the kinematic constraints must account for them.

14.3.3 Dynamic Pipe

Finite Volume dynamic pipe.

Definitions

The dynamic pipe is formulated according to the finite volume approach. The pipe is discretized by means of the pressures and the flow at the two ends, which are interpolated linearly. The mass and the momentum balance equations are written by cutting the pipe in two halves and adding the contribution of each resulting subvolume to the respective nodal equations.

Consider the mass conservation and the momentum balance equations for a one-dimensional flow:

$$\frac{D}{Dt} (dm) = 0, \quad (14.4)$$

$$\frac{D}{Dt} (dQ) = df. \quad (14.5)$$

When a rigid pipe is considered, the total derivative D/Dt of the test mass $dm = \rho A dx$ of Equation (14.4) yields

$$\frac{D}{Dt} (dm) = \frac{\partial}{\partial t} (dm) + v \frac{\partial}{\partial x} (dm), \quad (14.6)$$

which results in

$$q/x + A\rho/t = 0, \quad (14.7)$$

where $q = \rho Av$ is the mass flow. Consider now the momentum equation (14.5); the total derivative of the momentum $dQ = vdm$ yields

$$\frac{D}{Dt} (dQ) = (q/t + (qv)/x) dx, \quad (14.8)$$

while the pressure gradient and the viscous contributions can be isolated from the force per unit length on the right hand side:

$$df = -Adp + f_v dx + df^*, \quad (14.9)$$

so, by neglecting the deformability of the pipe and the extra forces df^* acting on the fluid, the momentum balance equation yields

$$q/t + (qv + Ap)/x = f_v, \quad (14.10)$$

which can be reduced to the pressure and flow unknowns simply by recalling the definition of the flow:

$$q/t + \left(\frac{q^2}{\rho A} + Ap \right)_{/x} = f_v. \quad (14.11)$$

A flexible pipe has been considered as well; the formulation is not reported for simplicity, because such a level of detail is required only for very specialized problems, and a first approximation can be obtained by altering the bulk modulus of the fluid.

The pipe is discretized by considering a finite volume approach, based on the use of constant stepwise (*Heavyside*) test functions with arbitrary trial functions. In the present case, linear trial functions have been considered both for the flow and for the pressure:

$$q(x) = \begin{bmatrix} 1-\xi & 1+\xi \\ 2 & 2 \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix},$$

$$p(x) = \begin{bmatrix} 1-\xi & 1+\xi \\ 2 & 2 \end{bmatrix} \begin{Bmatrix} p_1 \\ p_2 \end{Bmatrix},$$

with $\xi = \xi(x) \in [-1, 1]$ and $d\xi/dx = 2/(b-a)$. The discrete form of the pipe equations results in

$$\begin{aligned} q(b) - q(a) &= - \int_a^b \frac{\partial \rho}{\partial p} p_{/t} dx, \\ \frac{b-a}{2} \left(q(b)_{/t} + q(a)_{/t} \right) + \left(\frac{q(b)^2}{\rho(b) A} + Ap(b) \right) \\ - \left(\frac{q(a)^2}{\rho(a) A} + Ap(a) \right) &= \int_a^b f_v dx; \end{aligned}$$

by dividing the pipe in two portions, and by considering the domains $[-1, 0]$ and $[0, 1]$ for ξ in each portion, the discrete equations of the finite volume pipe become

$$\begin{aligned} -\frac{1}{2} (q_1 + q_2) - \frac{\partial \rho(-1/2)}{\partial p} \frac{L}{8} (3\dot{p}_1 + \dot{p}_2) &= \phi_1, \\ \frac{1}{2} (q_1 + q_2) - \frac{\partial \rho(1/2)}{\partial p} \frac{L}{8} (\dot{p}_1 + 3\dot{p}_2) &= \phi_2, \\ \frac{L}{8} (3\dot{q}_1 + \dot{q}_2) + \frac{(q_1 + q_2)^2}{4\rho(0) A} - \frac{q_1^2}{\rho(-1) A} \\ + \frac{A}{2} (p_2 - p_1) &= \frac{L}{2} \int_{-1}^0 f_v d\xi, \\ \frac{L}{8} (\dot{q}_1 + 3\dot{q}_2) + \frac{q_2^2}{\rho(1) A} - \frac{(q_1 + q_2)^2}{4\rho(0) A} \\ + \frac{A}{2} (p_2 - p_1) &= \frac{L}{2} \int_0^1 f_v d\xi, \end{aligned}$$

where ϕ_1 and ϕ_2 are the contributions of the two portions of pipe to the respective nodal flow balance equations. The integral of the time derivative of the density is numerically computed. The integral of the viscous forces per unit length is numerically performed as well, accounting for the flow regime in the pipe as function of the *Reynolds* number. In fact, for the forces per unit length, the dependency on the flow is considered linear for $0 < Re < 2000$, and quadratic for $Re > 4000$, while a polynomial fitting of the transition behavior, accounting also for the rate of the *Reynolds* number, is modeled for $2000 < Re < 4000$.

Equations

$$\begin{aligned} 0 &= \frac{1}{2} (q_1 + q_2) + AL \frac{\partial \rho}{\partial p_1} \left(\frac{3}{8} \dot{P}_1 + \frac{1}{8} \dot{P}_2 \right) \\ 0 &= -\frac{1}{2} (q_1 + q_2) + AL \frac{\partial \rho}{\partial p_2} \left(\frac{1}{8} \dot{P}_1 + \frac{3}{8} \dot{P}_2 \right) \\ 0 &= -L \left(\frac{3}{8} \dot{q}_1 + \frac{1}{8} \dot{q}_2 \right) - \left(\frac{1}{2} (q_1 + q_2) \right)^2 \frac{1}{\rho_m A} + q_1^2 \frac{1}{\rho_1 A} - \frac{A}{2} (p_2 - p_1) - f_1 \\ 0 &= -L \left(\frac{1}{8} \dot{q}_1 + \frac{3}{8} \dot{q}_2 \right) - q_2^2 \frac{1}{\rho_2 A} + \left(\frac{1}{2} (q_1 + q_2) \right)^2 \frac{1}{\rho_m A} - \frac{A}{2} (p_2 - p_1) - f_2 \\ p_1 &= P_1 \\ p_2 &= P_2 \end{aligned}$$

14.3.4 Pressure Flow Control Valve

$$R_1 = Q_{12} + Q_{13} \quad (14.12)$$

$$R_2 = -Q_{12} + Q_{24} \quad (14.13)$$

$$R_3 = -Q_{13} + Q_{34} \quad (14.14)$$

$$R_4 = -Q_{34} - Q_{24} \quad (14.15)$$

$$R_5 = A_v s; \quad (14.16)$$

$$R_6 = -A_v s \quad (14.17)$$

$$R_7 = -F + M\dot{v} + C\dot{s} + Ks + c_1 s + c_2 \dot{s} + c_3 \dot{v} + cf_1(s - s_{\max}) + cf_2 \dot{s} + cf_3 \dot{v} \quad (14.18)$$

$$R_8 = \dot{s} - v \quad (14.19)$$

Bibliography

- [1] Richard L. Bielawa. *Rotary Wing Structural Dynamics and Aeroelasticity*. AIAA, Washington, DC, 1992.
- [2] J. Gordon Leishman. *Principles of helicopter aerodynamics*. Cambridge University Press, 2 edition, 2006.
- [3] Huimin Zhang, Runsen Zhang, Yufeng Xing, and Pierangelo Masarati. On the optimization of n-sub-step composite time integration methods. *Nonlinear Dynamics*, 102(3):1939–1962, November 2020.
- [4] Huimin Zhang, Runsen Zhang, Andrea Zanoni, and Pierangelo Masarati. Performance of implicit A-stable time integration methods for multibody system dynamics. *Multibody System Dynamics*, 54(3):263–301, March 2022.
- [5] Pierangelo Masarati and Marco Morandini. An ideal homokinetic joint formulation for general-purpose multibody real-time simulation. *Multibody System Dynamics*, 20(3):251–270, October 2008. doi:10.1007/s11044-008-9112-8.
- [6] Pierangelo Masarati and Marco Morandini. Intrinsic deformable joints. To be published.
- [7] I. C. Cheeseman and N. E. Bennett. The effect of ground on a helicopter rotor in forward flight. TR 3021, NASA, 1955.
- [8] A. R. S. Bramwell. *Helicopter Dynamics*. Edward Arnold, London, 1976.
- [9] Dale M. Pitt and David A. Peters. Theoretical prediction of dynamic-inflow derivatives. *Vertica*, 5:21–34, 1981.

Appendix A

On the optimization of n -sub-step composite time integration methods

This appendix includes a verbatim copy of the paper “On the optimization of n -sub-step composite time integration methods”, written by Huimin Zhang, Runsen Zhang, Yufeng Xing and Pierangelo Masarati and published in Nonlinear Dynamics.

The paper reports some of MBDyn’s integrators coefficients.

The paper is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <https://creativecommons.org/licenses/by/4.0/>



On the optimization of n -sub-step composite time integration methods

Huimin Zhang · Runsen Zhang ·
Yufeng Xing · Pierangelo Maserati

Received: 12 May 2020 / Accepted: 12 October 2020 / Published online: 24 October 2020
© The Author(s) 2020

Abstract A family of n -sub-step composite time integration methods, which employs the trapezoidal rule in the first $n - 1$ sub-steps and a general formula in the last one, is discussed in this paper. A universal approach to optimize the parameters is provided for any cases of $n \geq 2$, and two optimal sub-families of the method are given for different purposes. From linear analysis, the first sub-family can achieve n th-order accuracy and unconditional stability with controllable algorithmic dissipation, so it is recommended for high-accuracy purposes. The second sub-family has second-order accuracy, unconditional stability with controllable algorithmic dissipation, and it is designed for heuristic energy-conserving purposes, by preserving as much low-frequency content as possible. Finally, some illustrative examples are solved to check the performance in linear and nonlinear systems.

Keywords n -Sub-step composite method · Optimization · High-accuracy · Energy-conserving

1 Introduction

Direct time integration methods are frequently used to predict accurate numerical responses for general dynamic problems after spatial discretization. Driven by the pursuit of desirable properties, including higher accuracy and efficiency, robust stability, and many others, a number of excellent methods were proposed in the past decades.

In terms of the formulations, existing methods are generally classified into explicit and implicit schemes. Explicit methods are mostly used in wave propagation problems, as their conditional stability limits the allowable time step size to the highest system frequency. Implicit methods have fewer restrictions on the problems to be solved due to the unconditional stability, but they require more computational efforts per step.

In another way, the integration methods can also be divided into single-step, multi-sub-step and multi-step techniques. The single-step methods only adopt the states of the last step to predict the current one, while the multi-sub-step methods also need the states at the intermediate collocation points, and the multi-step methods require the states of more than one previous step. Each of them has specific advantages and disadvantages.

From the literature, representative single-step methods include the Newmark method [25], the HHT- α method (by Hilbert, Hughes, and Taylor) [17], the WBZ- α method (by Wood, Bossak, and Zienkiewicz) [29], the generalized- α method [9], the GSSS (gener-

Huimin Zhang · Runsen Zhang · Yufeng Xing
School of Aeronautic Science and Engineering, Beihang
University, Beijing 100083, China

Huimin Zhang (✉) · Runsen Zhang · Pierangelo Maserati
Dipartimento di Scienze e Tecnologie Aerospaziali,
Politecnico di Milano, 20156 Milan, Italy
e-mail: huimin.zhang@polimi.it

alized single-step single-solve) method [34], and many others [28]. These single-step methods were proved to be spectrally identical to the linear multi-step methods [34], so they suffer from the Dahlquist's barrier [10], which states that the methods of higher than second-order accuracy cannot be unconditionally stable. Therefore, the methods mentioned above are all second-order accurate and unconditionally stable; some of them can also provide controllable algorithmic dissipation.

In the multi-step class, the Dahlquist's barrier certainly works, but in terms of accuracy, the linear two-step method [24, 33] is superior to most existing single-step methods under the same degree of algorithmic dissipation. In this class, BDFs (backward differentiation formulas) [11, 16] also represent a widely-used branch, particularly useful for stiff problems owing to the strong algorithmic dissipation. These popular multi-step methods are also second-order accurate and unconditionally stable. However, the multi-step methods are not self-starting, so another method has to be also used to solve the initial steps, which makes the multi-step methods not as convenient to use as the single-step ones.

The multi-sub-step methods, also known as multi-stage methods, allow more possibilities in terms of properties. The most representative method is the famous Runge–Kutta family [6, 7, 19], which can be designed to be arbitrarily higher-order accurate and unconditionally stable by choosing proper parameters and enough stages. Besides, Fung [12–15] provided some methods to reproduce the generalized Padé approximation. These methods can reach up to $2n$ th-order accuracy by employing n sampling grid points per step, but the dimension of the implicit equation to be solved is n times that of the original, resulting in huge computational costs. In the multi-sub-step class, the composite methods [3], which divide each step into several sub-steps and employ different methods in each sub-step, have received a lot of attention in recent years.

Based on Bank et al.'s work [1], Bathe et al. [3] introduced the concept of the n -sub-step composite method by utilizing the trapezoidal rule in the first $n - 1$ sub-steps and the $(n + 1)$ -point backward difference scheme at the end of the step. The two-sub-step scheme is known as the Bathe method, which is asymptotically stable with second-order accuracy. Thanks to its strong dissipation and preferable accuracy, the Bathe method has been found to perform well in many fields [2, 4, 27]. The three-, and four-sub-

step composite methods [8, 32], which are asymptotically stable with higher accuracy, were also developed adopting the similar idea. Furthermore, to acquire controllable algorithmic dissipation, the two-sub-step methods [20, 21, 26], and the controllable three-sub-step methods [18, 23], were proposed by replacing the backward difference scheme with a more general formula. However, with the increase in the number of sub-steps, the number of scalar parameters required to be designed also increases, so the basic requirements, including second-order accuracy, unconditional stability, controllable algorithmic dissipation, are not enough to determine these parameters uniquely. Two optimal sub-families of the controllable three-sub-step method were proposed in [23], since different conditions are considered as a supplement.

On this basis, this paper purposes to provide a universal approach to optimize the parameters of generalized n -sub-step composite method, where n can be any integer greater than 2, and the trapezoidal rule is employed in the first $n - 1$ sub-steps. Two kinds of optimization goals are considered, producing two optimal sub-families for different purposes. The first one intends to achieve higher-order accuracy, under the premises of unconditional stability and controllable algorithmic dissipation. The second one is dedicated to conserving low-frequency behavior, while still providing controllable high-frequency dissipation. From linear analysis, the resulting schemes in the first sub-family can reach up to n th-order accuracy by using n sub-steps, and the schemes in the second sub-family exhibit very small algorithmic dissipation in the low-frequency domain. Most of these schemes are developed for the first time, and in each sub-family, the accuracy can be improved by using more sub-steps. Finally, the proposed methods are applied to solve several numerical examples to check the performance.

This paper is organized as follows. The formulations of the n -sub-step composite method are shown in Sect. 2. The optimization of the parameters is implemented in Sect. 3. The detailed properties of the two sub-families are discussed in Sect. 4. Numerical examples are provided in Sect. 5, and conclusions are drawn in Sect. 6.

2 Formulation

In the literature, the composite methods were mostly developed to solve the problems in structural dynamics, as

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, t) = \mathbf{0}, \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad \dot{\mathbf{x}}(t_0) = \mathbf{v}_0 \quad (1)$$

where \mathbf{M} is the mass matrix, \mathbf{F} collects the damping force, internal force and external load, \mathbf{x} , $\dot{\mathbf{x}}$ and $\ddot{\mathbf{x}}$ are the displacement, velocity and acceleration vectors, respectively, t is the time, and t_0 , \mathbf{x}_0 and \mathbf{v}_0 are the given initial time, displacement and velocity, respectively. When this method is applied using n sub-steps, it can be formulated as

$$\mathbf{M}\ddot{\mathbf{x}}_{k+2j\gamma} + \mathbf{F}(\mathbf{x}_{k+2j\gamma}, \dot{\mathbf{x}}_{k+2j\gamma}, t_k + 2j\gamma h) = \mathbf{0} \quad (2a)$$

$$\mathbf{x}_{k+2j\gamma} = \mathbf{x}_{k+2(j-1)\gamma} + \gamma h (\dot{\mathbf{x}}_{k+2(j-1)\gamma} + \dot{\mathbf{x}}_{k+2j\gamma}) \quad (2b)$$

$$\dot{\mathbf{x}}_{k+2j\gamma} = \dot{\mathbf{x}}_{k+2(j-1)\gamma} + \gamma h (\ddot{\mathbf{x}}_{k+2(j-1)\gamma} + \ddot{\mathbf{x}}_{k+2j\gamma}) \quad (2c)$$

$$j = 1, 2, 3, \dots, n - 1 \quad (2d)$$

and

$$\mathbf{M}\ddot{\mathbf{x}}_{k+1} + \mathbf{F}(\mathbf{x}_{k+1}, \dot{\mathbf{x}}_{k+1}, t_k + h) = \mathbf{0} \quad (3a)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h \left(\sum_{j=0}^{n-1} q_j \dot{\mathbf{x}}_{k+2j\gamma} + q_n \dot{\mathbf{x}}_{k+1} \right) \quad (3b)$$

$$\dot{\mathbf{x}}_{k+1} = \dot{\mathbf{x}}_k + h \left(\sum_{j=0}^{n-1} q_j \ddot{\mathbf{x}}_{k+2j\gamma} + q_n \ddot{\mathbf{x}}_{k+1} \right) \quad (3c)$$

where $\mathbf{x}_k \approx \mathbf{x}(t_k)$ is the numerical solution at step k , $\mathbf{x}_{k+2j\gamma} \approx \mathbf{x}(t_k + 2j\gamma h)$ ($j = 1, 2, \dots, n - 1$) denotes the numerical solution at collocation points, h is the step size, and $\gamma, q_0, q_1, \dots, q_n$ are the control parameters. The current step $[t_k, t_k + h]$ is divided into n sub-steps: $[t_k, t_k + 2\gamma h], [t_k + 2\gamma h, t_k + 4\gamma h], \dots, [t_k + 2(n-2)\gamma h, t_k + 2(n-1)\gamma h]$, and $[t_k + 2(n-1)\gamma h, t_k + h]$. In the first $n - 1$ sub-steps, the trapezoidal rule is adopted. In the last one, a general formula containing information about all collocation points is utilized. The present formulation can reduce to the ρ_∞ -Bathe method [26] when $n = 2$ and to the three-sub-step method [18, 23] when $n = 3$.

In this method, because the same form of assumptions is used to solve \mathbf{x}_{k+1} and $\dot{\mathbf{x}}_{k+1}$, Eqs. (2) and (3)

can be reformulated based on the general first-order differential equation $f(\mathbf{y}, \dot{\mathbf{y}}, t) = \mathbf{0}$, as

$$f(\mathbf{y}_{k+2j\gamma}, \dot{\mathbf{y}}_{k+2j\gamma}, t_k + 2j\gamma h) = \mathbf{0} \quad (4a)$$

$$\mathbf{y}_{k+2j\gamma} = \mathbf{y}_{k+2(j-1)\gamma} + \gamma h (\dot{\mathbf{y}}_{k+2(j-1)\gamma} + \dot{\mathbf{y}}_{k+2j\gamma}) \quad (4b)$$

$$j = 1, 2, 3, \dots, n - 1 \quad (4c)$$

and

$$f(\mathbf{y}_{k+1}, \dot{\mathbf{y}}_{k+1}, t_k + h) = \mathbf{0} \quad (5a)$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h \left(\sum_{j=0}^{n-1} q_j \dot{\mathbf{y}}_{k+2j\gamma} + q_n \dot{\mathbf{y}}_{k+1} \right) \quad (5b)$$

where $\{\mathbf{x}; \dot{\mathbf{x}}\}$ is replaced by \mathbf{y} , and the dynamics equations can be equivalently formulated as first-order differential equations by adding the trivial equation $\dot{\mathbf{x}} = \dot{\mathbf{x}}$. Equations (4) and (5) present more general formulations for solving first-order and arbitrarily higher-order differential equations. However, for solving the second-order dynamic problems, Eqs. (2) and (3) are still more recommended, since in the equivalent first-order expressions, the number of implicit equations to be solved doubles.

From the formulation, the first $n - 1$ sub-steps can share the same procedure in a loop, whereas the last sub-step needs to be implemented separately. The assumption $q_n = \gamma$ is introduced here, which imposes that the last sub-step shares the same form of Jacobi matrix as the first $n - 1$ sub-steps. This assumption is particularly useful when applied to linear problems, since it allows the constant Jacobi matrix to be factorized only once, like in the single-step methods. For applications, Table 1 shows the computational procedures of the n -sub-step composite method for the general first-order differential equation $f(\mathbf{y}, \dot{\mathbf{y}}, t) = \mathbf{0}$, where the Newton-Raphson iteration is utilized to solve the nonlinear equation per sub-step.

Besides, by reorganizing the formulations, the composite method can be regarded as a special case of the diagonally-implicit Runge–Kutta methods (DIRKs) with the explicit first-stage. The corresponding Butcher's tableau [6] has the form as

Table 1 Computational procedure of the n -sub-step composite method for solving $f(y, \dot{y}, t) = \mathbf{0}$, $y(t_0) = y_0$ **A. Initial calculations**

1. From the function $f(y, \dot{y}, t)$ and its derivative functions with respect to y and \dot{y} , as f_y and $f_{\dot{y}}$, respectively;
2. Initialize t_0 , y_0 and \dot{y}_0 ;
3. Select the time step size h , the algorithmic parameters $\gamma, q_0, q_1, \dots, q_{n-1}$, the tolerance error ϵ , and the maximum number of iterations N ;
4. Calculate the constant: $a = \frac{1}{\gamma h}$.

B. For each time step

1. The first $n - 1$ sub-steps

For $j = 1, j < n, j++$:

- a. Predict $y_{k+2j\gamma}$ and $\dot{y}_{k+2j\gamma}$:

$$i = 0, y_{k+2j\gamma} = y_{k+2(j-1)\gamma} + 2\gamma h \dot{y}_{k+2(j-1)\gamma}, \dot{y}_{k+2j\gamma} = a(y_{k+2j\gamma} - y_{k+2(j-1)\gamma}) - \dot{y}_{k+2(j-1)\gamma};$$

- b. Prepare the matrices:

$$f_{k+2j\gamma} = f(y_{k+2j\gamma}, \dot{y}_{k+2j\gamma}, t_k + 2j\gamma h), f_{y,k+2j\gamma} = f_y(y_{k+2j\gamma}, \dot{y}_{k+2j\gamma}, t_k + 2j\gamma h), f_{\dot{y},k+2j\gamma} = f_{\dot{y}}(y_{k+2j\gamma}, \dot{y}_{k+2j\gamma}, t_k + 2j\gamma h);$$

- c. Update $y_{k+2j\gamma}$ and $\dot{y}_{k+2j\gamma}$:

$$i = i + 1, \Delta y_{k+2j\gamma} = -(f_{y,k+2j\gamma} + a f_{\dot{y},k+2j\gamma})^{-1} f_{k+2j\gamma}, y_{k+2j\gamma} = y_{k+2j\gamma} + \Delta y_{k+2j\gamma}, \dot{y}_{k+2j\gamma} = \dot{y}_{k+2j\gamma} + a \Delta y_{k+2j\gamma};$$

- d. If $i < N$ and $|f_{k+2j\gamma}| > \epsilon$, go to b; If $i \equiv N$ and $|f_{k+2j\gamma}| > \epsilon$, abort.

End.

2. The last sub-step

- a. Predict y_{k+1} and \dot{y}_{k+1} :

$$i = 0, y_{k+1} = y_{k+2(n-1)\gamma} + (1 - 2(n-1)\gamma)h \dot{y}_{k+2(n-1)\gamma}, \dot{y}_{k+1} = a(y_{k+1} - y_k - h \sum_{j=0}^{n-1} q_j \dot{y}_{k+2j\gamma});$$

- b. Prepare the matrices:

$$f_{k+1} = f(y_{k+1}, \dot{y}_{k+1}, t_k + h), f_{y,k+1} = f_y(y_{k+1}, \dot{y}_{k+1}, t_k + h), f_{\dot{y},k+1} = f_{\dot{y}}(y_{k+1}, \dot{y}_{k+1}, t_k + h);$$

- c. Update y_{k+1} and \dot{y}_{k+1} :

$$i = i + 1, \Delta y_{k+1} = -(f_{y,k+1} + a f_{\dot{y},k+1})^{-1} f_{k+1}, y_{k+1} = y_{k+1} + \Delta y_{k+1}, \dot{y}_{k+1} = \dot{y}_{k+1} + a \Delta y_{k+1};$$

- d. If $i < N$ and $|f_{k+1}| > \epsilon$, go to b; If $i \equiv N$ and $|f_{k+1}| > \epsilon$, abort.

$$\begin{array}{c|cccccc} 0 & 0 & 0 & \cdots & 0 & 0 \\ 2\gamma & \gamma & \gamma & 0 & \cdots & 0 & 0 \\ 4\gamma & \gamma & 2\gamma & \gamma & \ddots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 2(n-1)\gamma & \gamma & 2\gamma & 2\gamma & \cdots & \gamma & 0 \\ 1 & q_0 & q_1 & q_2 & \cdots & q_{n-1} & \gamma \\ \hline & q_0 & q_1 & q_2 & \cdots & q_{n-1} & \gamma \end{array}$$

order differential equation is discussed, as

$$\dot{y} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\xi\omega \end{bmatrix} y, \quad y = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \quad (7)$$

Decomposing the coefficient matrix in Eq. (7) yields the simplified first-order equation

$$\dot{y} = \lambda y, \quad \lambda = (-\xi \pm i\sqrt{1-\xi^2})\omega \quad (8)$$

where $i = \sqrt{-1}$. When the composite method is applied, the recursive scheme becomes

$$y_{k+1} = A(\lambda h) y_k \quad (9)$$

3 Optimization

In linear spectral analysis, owing to the mode superposition principle, it is common and enough to consider the single degree-of-freedom equation

$$\ddot{x} + 2\xi\omega\dot{x} + \omega^2 x = 0 \quad (6)$$

where ξ is the damping ratio, and ω is the natural frequency. To simplify the analysis, the equivalent first-

where the amplification factor A is

$$A(z) = (1 - q_n z)^{-1} \left(1 + z \sum_{j=0}^{n-1} q_j \left(\frac{1 + \gamma z}{1 - \gamma z} \right)^j \right), \\ z = \lambda h \quad (10)$$

Since $q_n = \gamma$ is assumed in Sect. 2, Eq. (10) is updated as

$$A(z) = \frac{(1 - \gamma z)^{n-1} + z \sum_{j=0}^{n-1} (q_j (1 + \gamma z)^j (1 - \gamma z)^{n-j-1})}{(1 - \gamma z)^n} \\ = \frac{1 + a_1 z + a_2 z^2 + \cdots + a_n z^n}{(1 - \gamma z)^n} \quad (11)$$

where the coefficient of z^p ($p = 1, 2, \dots, n$) is represented by a_p ($p = 1, 2, \dots, n$), expressed as

$$a_p = \binom{p}{n-1} (-\gamma)^p \\ + \gamma^{p-1} \sum_{j=0}^{n-1} \left(q_j \sum_{m=\max\{0, p+j-n\}}^{\min\{j, p-1\}} P(m, j, p, n) \right), \\ P(m, j, p, n) = (-1)^{p-m-1} \binom{m}{j} \binom{p-m-1}{n-j-1}, \\ p = 1, 2, \dots, n-1 \quad (12)$$

and

$$a_n = \gamma^{n-1} \sum_{j=0}^{n-1} ((-1)^{n-j-1} q_j) \quad (13)$$

For example, $n = 5$ follows

$$a_1 = -4\gamma + q_0 + q_1 + q_2 + q_3 + q_4 \quad (14a)$$

$$a_2 = 6\gamma^2 + \gamma(-4q_0 - 2q_1 + 2q_3 + 4q_4) \quad (14b)$$

$$a_3 = -4\gamma^3 + \gamma^2(6q_0 - 2q_2 + 6q_4) \quad (14c)$$

$$a_4 = \gamma^4 + \gamma^3(-4q_0 + 2q_1 - 2q_3 + 4q_4) \quad (14d)$$

$$a_5 = \gamma^4(q_0 - q_1 + q_2 - q_3 + q_4) \quad (14e)$$

Consequently, the parameters under analysis change from q_j ($j = 0, 1, \dots, n-1$) and γ , to a_p ($p = 1, 2, \dots, n$) and γ in the following. When a_p and γ are given, the parameters q_j can be obtained uniquely by solving Eqs. (12) and (13). For applications, Table 2 shows

Table 2 Formulas of q_j ($j = 0, 1, \dots, n-1$)

n	q_j ($j = 0, 1, \dots, n-1$)
2	$q_0 = \frac{\gamma}{2} + \frac{a_1}{2} - \frac{a_2}{2\gamma}$
	$q_1 = \frac{\gamma}{2} + \frac{a_1}{2} + \frac{a_2}{2\gamma}$
3	$q_0 = \frac{3\gamma}{4} + \frac{a_1}{4} - \frac{a_2}{4\gamma} + \frac{a_3}{4\gamma^2}$
	$q_1 = \gamma + \frac{a_1}{2} - \frac{a_3}{2\gamma^2}$
4	$q_2 = \frac{\gamma}{4} + \frac{a_1}{4} + \frac{a_2}{4\gamma} + \frac{a_3}{4\gamma^2}$
	$q_0 = \frac{7\gamma}{8} + \frac{a_1}{8} - \frac{a_2}{8\gamma} + \frac{a_3}{8\gamma^2} - \frac{a_4}{8\gamma^3}$
5	$q_1 = \frac{11\gamma}{8} + \frac{3a_1}{8} - \frac{a_2}{8\gamma} - \frac{a_3}{8\gamma^2} + \frac{3a_4}{8\gamma^3}$
	$q_2 = \frac{5\gamma}{8} + \frac{3a_1}{8} + \frac{a_2}{8\gamma} - \frac{a_3}{8\gamma^2} - \frac{3a_4}{8\gamma^3}$
	$q_3 = \frac{\gamma}{8} + \frac{a_1}{8} + \frac{a_2}{8\gamma} + \frac{a_3}{8\gamma^2} + \frac{a_4}{8\gamma^3}$
	$q_0 = \frac{15\gamma}{16} + \frac{a_1}{16} - \frac{a_2}{16\gamma} + \frac{a_3}{16\gamma^2} - \frac{a_4}{16\gamma^3} + \frac{a_5}{16\gamma^4}$
	$q_1 = \frac{13\gamma}{8} + \frac{a_1}{4} - \frac{a_2}{8\gamma} + \frac{a_4}{8\gamma^3} - \frac{a_5}{4\gamma^4}$
	$q_2 = \gamma + \frac{3a_1}{8} - \frac{a_3}{8\gamma^2} + \frac{3a_5}{8\gamma^4}$
	$q_3 = \frac{3\gamma}{8} + \frac{a_1}{4} + \frac{a_2}{8\gamma} - \frac{a_4}{8\gamma^3} - \frac{a_5}{4\gamma^4}$
	$q_4 = \frac{\gamma}{16} + \frac{a_1}{16} + \frac{a_2}{16\gamma} + \frac{a_3}{16\gamma^2} + \frac{a_4}{16\gamma^3} + \frac{a_5}{16\gamma^4}$

the formulas of q_j expressed by a_p and γ for the cases $n = 2, 3, 4, 5$.

3.1 Higher-order schemes

A numerical method is naturally expected to be as accurate as possible, so the higher-order schemes are considered first. From the scheme of Eq. (9), the composite method uses the amplification factor A , rewritten as

$$A(z) = \frac{1 + a_1 z + a_2 z^2 + \cdots + a_n z^n}{(1 - \gamma z)^n} \quad (15)$$

to approximate the exact amplification factor \hat{A}

$$\hat{A}(z) = e^z = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \dots \quad (16)$$

Hence the local truncation error σ can be defined as

$$\sigma = y_{k+1} - y(t_{k+1}) = (A(z) - \hat{A}(z))y(t_k) \quad (17)$$

If $\sigma = O(z^{s+1})$, the method is said to be s th-order accurate, which requires that up to s th derivatives of A at $z = 0$ are all equal to 1, that is

$$A(0) = A^{(1)}(0) = A^{(2)}(0) = \dots = A^{(s)}(0) = 1 \quad (18)$$

To satisfy Eq. (18), a_p ($p = 1, 2, \dots, n$) can be solved as

$$A^{(1)}(0) = 1 \Rightarrow a_1 = 1 - n\gamma \quad (19a)$$

$$A^{(2)}(0) = 1 \Rightarrow a_2 = \frac{1}{2} - n\gamma + \frac{n(n-1)}{2}\gamma^2 \quad (19b)$$

$$A^{(3)}(0) = 1 \Rightarrow \quad (19c)$$

$$a_3 = \frac{1}{6} - \frac{n}{2}\gamma + \frac{n(n-1)}{2}\gamma^2 - \frac{n(n-1)(n-2)}{6}\gamma^3 \quad (19d)$$

$$\dots \quad (19e)$$

Therefore, if all a_p ($p = 1, 2, \dots, n$) follow the relationships in Eq. (19), this method can achieve n th-order accuracy, and then γ becomes the only free parameter to control the stability.

A time integration method is said to be unconditionally stable if $|A(z)| \leq 1$ for all $\Re(z) \leq 0$ where $z = \lambda h = (-\xi \pm i\sqrt{1-\xi^2})wh$. According to Ref. [19], the bounds on γ can be given by considering the stability on the imaginary axis ($\xi = 0$), which can result in the unconditional stability when the accuracy order $s = n$ in the DIRKs. Therefore, let $z = \pm i\tau$ where $\tau = wh$ is a real number, and

$$N(z) = 1 + a_1 z + a_2 z^2 + \dots + a_n z^n \quad (20a)$$

$$D(z) = (1 - \gamma z)^n \quad (20b)$$

which are the numerator and denominator of $A(z)$ in Eq. (15), respectively, $|A(z)| \leq 1$ is equivalent to

$$|A(z)|^2 = A(i\tau)A(-i\tau) = \frac{N(i\tau)N(-i\tau)}{D(i\tau)D(-i\tau)} \leq 1 \quad (21)$$

Then the condition for unconditional stability can be transformed into

$$S(\tau) = D(i\tau)D(-i\tau) - N(i\tau)N(-i\tau) = \sum_{j=0}^n (c_{2j}\tau^{2j}) \geq 0 \text{ for } \tau \geq 0 \quad (22)$$

where the function $S(\tau)$ is introduced, and the coefficients c_{2j} ($j = 0, 1, 2, \dots, n$) are expressed as

$$c_{2j} = \binom{j}{n} \gamma^{2j} + (-1)^{j+1} \sum_{m=\max\{0,2j-n\}}^{\min\{n,2j\}} ((-1)^m a_m a_{2j-m}) \quad (23)$$

in which a_0 is set to 1. By Eq. (22), the bounds on γ of the cases $n = 2, 3, 4, 5$ are provided in Table 3. It follows that, with $s = n$, the allowable range of γ narrows as n increases and, in some cases, the n -sub-step method can achieve $(n+1)$ th-order accuracy with a fixed γ .

Besides, algorithmic dissipation is also a desirable property for a time integration method, to filter out the inaccurate high-frequency content. Generally, it is measured by the spectral radius ρ_∞ at high-frequency limit, that is

$$|A(z)| \rightarrow \rho_\infty \text{ as } |z| = \omega h \rightarrow +\infty, \rho_\infty \in [0, 1] \quad (24)$$

and it gets stronger with a smaller ρ_∞ . With $A(z)$ from Eq. (15), Eq. (24) can be satisfied if

$$a_n^2 = \left(\sum_{j=0}^n \left(\frac{(-1)^j}{(n-j)!} \binom{j}{n} \gamma^j \right) \right)^2 = \rho_\infty^2 \gamma^{2n} \quad (25)$$

which can be used to solve γ for a given ρ_∞ . Table 4 shows the solutions of γ for several specific ρ_∞ in

Table 3 Bounds on γ for unconditional stability (s is the accuracy order) in the higher-order schemes

n	s	Bounds on γ
2	2	[0.2500000000000000, $+\infty$]
	3	0.788675134594813
3	3	[0.3333333333333333, 1.068579021301628]
	4	1.068579021301628
4	4	[0.394337567297396, 1.280579761275305]
	5	—
5	5	[0.246505193142435, 0.361803398875471] \cup [0.420782512765729, 0.473268391258294]
	6	0.473268391258294

the cases $n = 2, 3, 4, 5$. Note that Eq. (25) has multiple solutions; the smallest one that meets the requirement of unconditional stability, as shown in Table 3, is selected.

So far, the unconditionally stable higher-order accurate schemes with controllable algorithmic dissipation have been developed, whose parameter γ can be solved for a given ρ_∞ by Eq. (25), a_p ($p = 1, 2, \dots, n$) are determined by γ as shown in Eq. (19), and then q_j ($j = 0, 1, 2, \dots, n - 1$) can be obtained by solving Eqs. (12) and (13). These information for the cases $n = 2, 3, 4, 5$ are shown in Tables 2, 3 and 4. The special case $n = 2$ is identical to the ρ_∞ -Bathe method [26], whereas the other cases are presented here for the first time. In addition, the accuracy and algorithmic dissipation are discussed in more detail in Sect. 4.

3.2 Conserving schemes

An original intention of the composite methods was to conserve the energy of the system [2], which explains why the trapezoidal rule is utilized in most sub-steps. Existing two- and three-sub-step methods [18, 22, 26] really show preferable energy-conserving characteristic over other single- and multi-step methods. In this work, a simple and general approach to determine the parameters, which enable the n -sub-step composite method to conserve as much low-frequency content as possible, is proposed.

First of all, to be competitive, the method needs to have some basically useful properties, including at least second-order accuracy, which requires

$$a_1 = 1 - n\gamma \quad (26a)$$

$$a_2 = \frac{1}{2} - n\gamma + \frac{n(n-1)}{2}\gamma^2 \quad (26b)$$

and controllable algorithmic dissipation, achieved by

$$a_n^2 = \rho_\infty^2 \gamma^{2n} \quad (27)$$

In addition, unconditional stability also needs to be satisfied, which will be checked last.

To conserve the energy as much as possible, the spectral radius $\rho = |A(z)|$ should be as close to 1 as possible over the low-frequency range. For the special case $\rho_\infty = 1$, ρ should remain 1 in the whole frequency domain. For other cases $0 \leq \rho_\infty < 1$, the departure of ρ from unit value should be as slow as possible from $\rho(0) = 1$. Considering the conservative system ($\xi = 0$), this purpose can be realized by making the function $S(\tau)$, defined in Eq. (22), as smooth as possible. It follows that $S(0) = S^{(1)}(0) = S^{(2)}(0) = \dots = S^{(m)}(0) = 0$, where $S^{(m)}(0)$ is the m th-order derivative of $S(\tau)$ at $\tau = 0$, and m should be as large as possible. As $S(\tau)$ is a linear polynomial, the condition transforms into its coefficients c_{2j} ($j = 0, 1, 2, \dots, m$) = 0. To clarify, c_{2j} ($j = 0, 1, 2, \dots, n$) are enumerated as

$$c_0 = 0 \quad (28a)$$

$$c_2 = n\gamma^2 - a_1^2 + 2a_0a_2 \quad (28b)$$

$$c_4 = \frac{n(n-1)}{2}\gamma^4 - a_2^2 + 2a_1a_3 - 2a_0a_4 \quad (28c)$$

$$c_6 = \frac{n(n-1)(n-2)}{6}\gamma^6 - a_3^2 + 2a_2a_4 - 2a_1a_5 \\ + 2a_0a_6 \quad (28d)$$

$$\dots \quad (28e)$$

$$c_{2n-2} = n\gamma^{2n-2} - a_{n-1}^2 + 2a_{n-2}a_n \quad (28f)$$

Table 4 γ for controllable algorithmic dissipation in the higher-order schemes

ρ_∞	$n = 2$	$n = 3$	$n = 4$	$n = 5$
0.0	0.292893218813452	0.435866521508460	0.572816062482135	0.278053841136450
0.1	0.287089056989371	0.421486815409409	0.548366644975830	0.274141306031868
0.2	0.281754163448146	0.408500789512922	0.526386456842386	0.270459886774582
0.3	0.276820321671636	0.396647209121134	0.506330118970782	0.266978043925651
0.4	0.272233289109874	0.385731000460835	0.487797474812348	0.263670231711606
0.5	0.267949192431123	0.375602225015285	0.470480577621677	0.260515416607055
0.6	0.263932022500210	0.366142810103347	0.454130785036529	0.257496029856675
0.7	0.260151847569038	0.357257811967234	0.438536189902193	0.254597208170133
0.8	0.256583509747431	0.348869453074869	0.423503766067179	0.251806231183850
0.9	0.253205655191036	0.340912922771929	0.408841866120699	0.249112096529630
1.0	0.250000000000000	0.333333333333333	0.394337567297407	0.246505193142820

$$c_{2n} = \gamma^{2n} - a_n^2 \quad (28g)$$

From Eqs. (26) and (27), we can obtain $c_2 = 0$ and $c_{2n} = (1 - \rho_\infty^2)\gamma^{2n} \geq 0$, respectively. For the case $n = 2$, the conditions on accuracy and algorithmic dissipation are enough to determine all parameters, resulting again in the ρ_∞ -Bathe method [26]. For other cases with $n > 2$, the $n - 2$ remaining parameters, a_3, a_4, \dots, a_{n-1} and γ , are obtained by solving the equations $c_4 = c_6 = \dots = c_{2n-2} = 0$. The values of these parameters for the cases $n = 3, 4, 5$ are shown in Table 5, where the set with γ close to $\frac{1}{2n}$ is selected, which requires $a_n = \rho_\infty \gamma^n$.

Then all parameters of the conserving schemes have been given by combining Eqs. (12), (13), (26), (27) and $c_4 = c_6 = \dots = c_{2n-2} = 0$. The resulting scheme of $n = 3$ is equivalent to the first sub-family of the three-sub-step method proposed in [23]; the other cases are presented here for the first time.

In particular, when $\rho_\infty = 1$, the resulting scheme is a n -sub-step method with the trapezoidal rule in all sub-steps, which is supposed to be unconditionally stable in the linear analysis. Empirically, the algorithmic dissipation is acquired by reducing the spectral radius ρ , so the dissipative schemes are likely to be also unconditionally stable, and even present more robust stability. For the undamped case ($\xi = 0$), the stability can be guaranteed since $S(\tau) = (1 - \rho_\infty^2)\gamma^{2n}\tau^{2n} \geq 0$; for other cases, the stability conditions of the schemes listed in Table 5 are checked one by one by considering $\xi \in (0, 1]$ and $\tau \in [0, 10000]$ numerically. As expected, $\rho \leq 1$ is satisfied at every point in

all schemes, so these methods can be said to possess unconditional stability for linear problems. Other properties are discussed in Sect. 4.

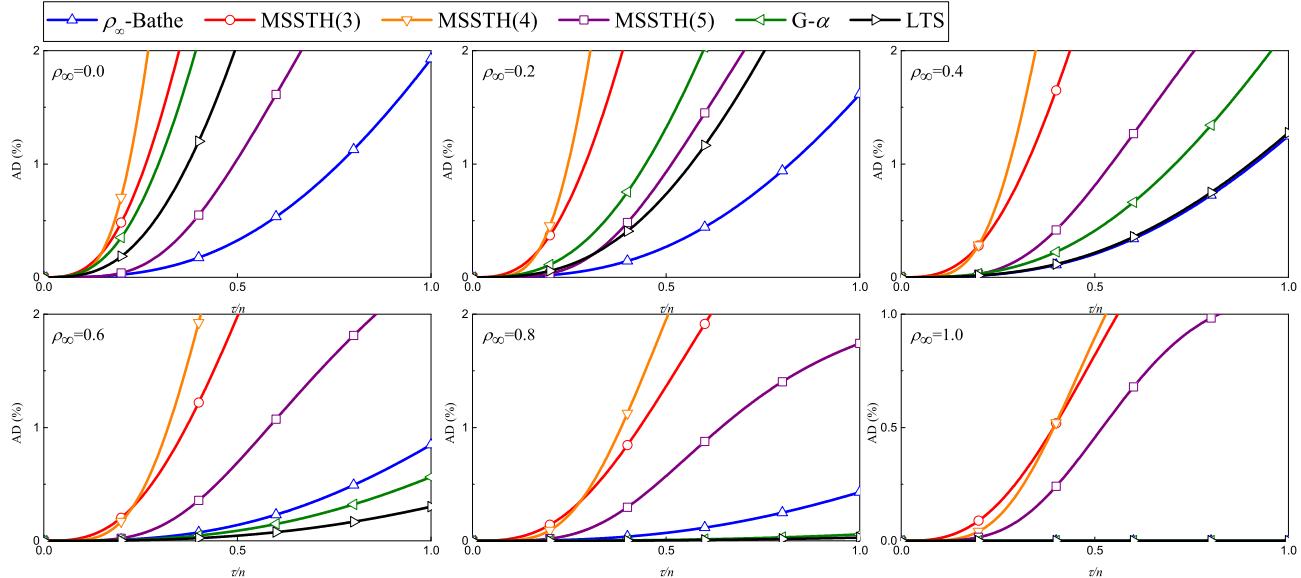
4 Properties

Two sub-families of the n -sub-step composite method have been presented for different purposes. To identify them, the higher-order schemes are referred to as MSSTH(n), and the conserving schemes are MSSTC(n), where MSST means the multi-sub-step composite method which employs the trapezoidal rule in all sub-steps except the last one, H and C are utilized to distinguish the two sub-families, and n is the number of sub-steps.

In this section, the representative methods in the literature, including the single-step generalized- α method [9] (G- α) and the linear two-step method [33] (LTS) are also considered for comparison. As the employed methods are all implicit, their computational cost is mainly spent on the iterative calculation when used for nonlinear problems, or the matrix factorization for linear problems. The vector operations brought by the recursive scheme of the method itself is generally considered to have little effect on overall efficiency. Therefore, G- α and LTS are recognized as having equivalent efficiency if the same step size is used. As the composite methods implement a single-step or multi-step scheme in each sub-step, they have the equivalent efficiency to G- α and LTS, if their required number of sub-steps is equal to the number of steps required by G- α and LTS. For this

Table 5 a_p ($p = 3, 4, \dots, n - 1$) and γ for controllable algorithmic dissipation in the conserving schemes

ρ_∞	$n = 3$		$n = 4$		$n = 5$		γ
	γ	a_3	γ	a_3	a_4	γ	
0.0	0.180425306429398	0.00453529185986996	0.131378736730466	0.00763819606391975	0.000257160742971488	0.103557108920215	
0.1	0.178619458204658	0.00494493283913114	0.130548620946472	0.00793598250555122	0.000286872844754286	0.103095631511675	
0.2	0.176945806618224	0.00532986673141648	0.12977758318848	0.00821506063336095	0.000314958374958848	0.102666675025093	
0.3	0.175385515842846	0.00569340893903897	0.129057207257355	0.00847799948547669	0.000341631205811411	0.102265594492185	
0.4	0.173923607877197	0.00603821324069812	0.128380804919945	0.00872684827484184	0.000367062789628561	0.101888703879882	
0.5	0.172547961422089	0.00636644119939074	0.127742970556848	0.00896327074168002	0.000391393000239752	0.101533025147874	
0.6	0.171248618590691	0.00667987987582935	0.127139265902084	0.00918863822587218	0.00041473766914113	0.101196115073181	
0.7	0.170017291772476	0.00698002584685393	0.126565999083137	0.00940409590940791	0.000437193952981663	0.100875942445807	
0.8	0.168847004679168	0.00726814657645952	0.126020063586496	0.00961061107396470	0.000458844285976519	0.100570798918745	
0.9	0.167731825756887	0.00754532615554921	0.125498818830422	0.00980900897660744	0.000479759288032603	0.100279232954742	
1.0	0.1666666666666667	0.00781250000000000	0.1250000000000000	0.0100000000000000	0.0005000000000000	0.1000000000000000	

**Fig. 1** Percentage amplitude decay for MSSTH(2,3,4,5), G- α and LTS

reason, to compare the properties under the close computational costs, the same h/n , where n is the number of sub-steps in the composite methods, and $n = 1$ for the G- α and LTS, is used in these methods.

As discussed in Sect. 3, MSSTH(n) has n th-order accuracy under the premises of unconditional stability and controllable algorithmic dissipation. Figures 1 and 2 display the percentage amplitude decay (AD(%)) and period elongation (PE(%)) respectively, of which the definition can refer to [34], of MSSTH(2,3,4,5), G- α and LTS, considering the undamped case ($\xi = 0$). The abscissa is set as τ/n to compare these methods under the close computational costs.

The results illustrate that the amplitude and period accuracy cannot be improved simultaneously as the order of accuracy increases in MSSTH(n). In terms of amplitude, with a small ρ_∞ , the ρ_∞ -Bathe method (the same as MSSTH(2)) is the most accurate, and when $0.4 < \rho_\infty \leq 1$, LTS shows smaller dissipation error, followed by the G- α and the ρ_∞ -Bathe method. From Fig. 2, MSSTH(3,4,5) have smaller period error than the second-order methods, and MSSTH(5) is the best among them.

In the same way, the percentage amplitude decay and period elongation of MSSTC(2,3,4,5), G- α and LTS for the undamped case are shown in Figs. 3 and 4,

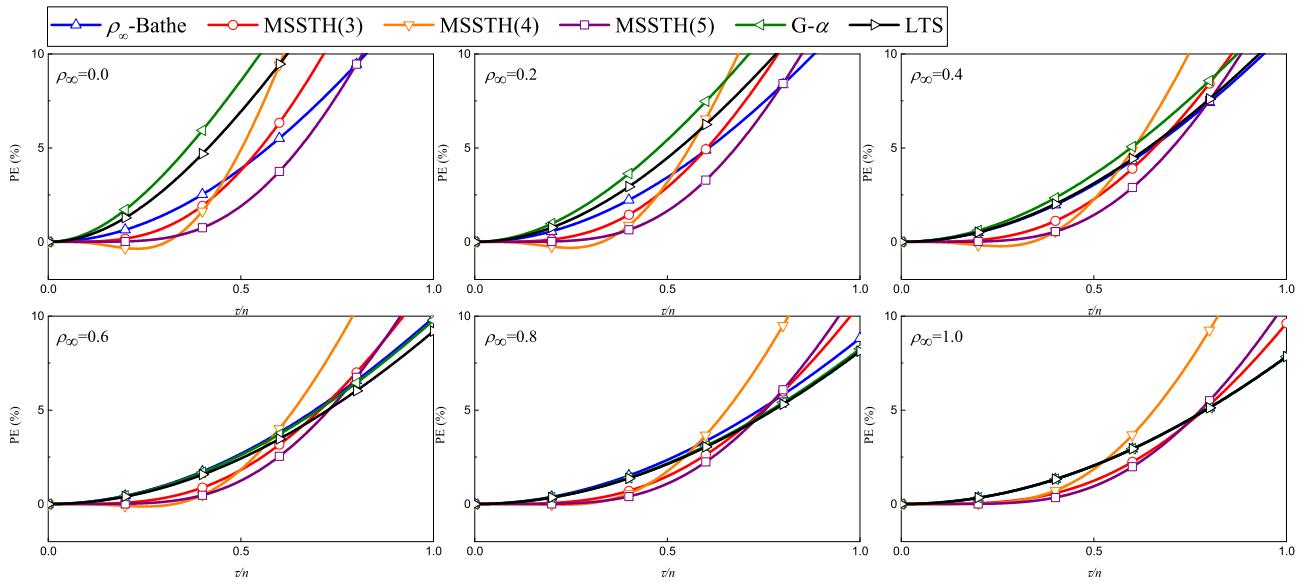


Fig. 2 Percentage period elongation for MSSTH(2,3,4,5), G- α and LTS

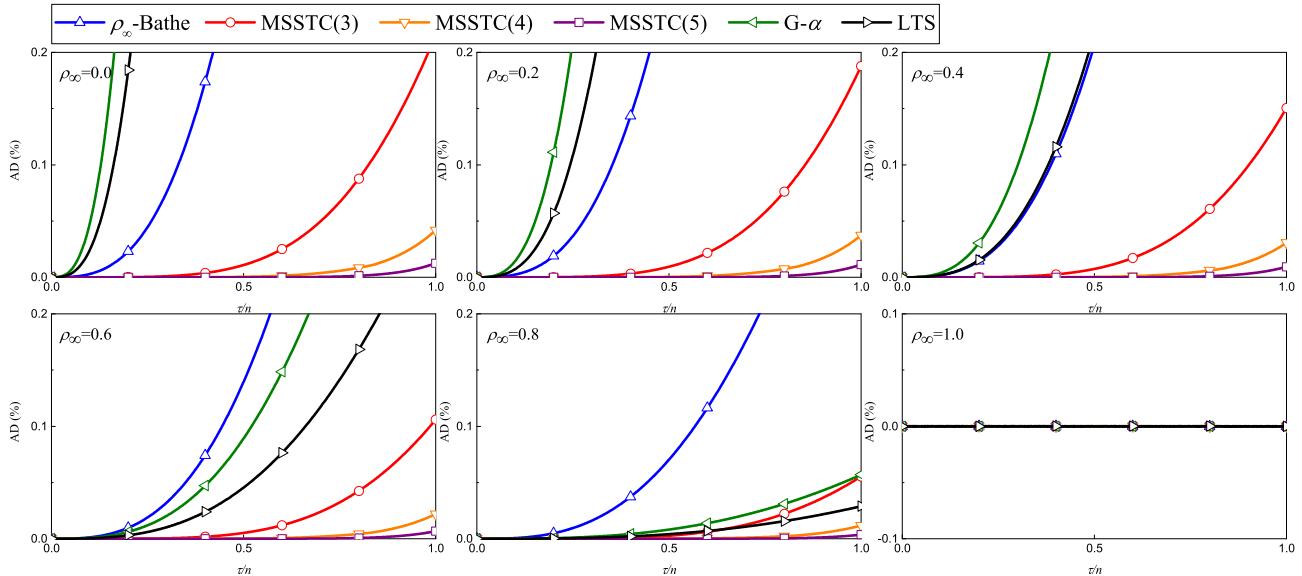


Fig. 3 Percentage amplitude decay for MSSTC(2,3,4,5), G- α and LTS

respectively. It can be observed that under the similar efficiency, MSSTC(n) presents higher amplitude and period accuracy with a larger n . The gap is more obvious as ρ_∞ decreases, and when $\rho_\infty = 1$, all the schemes have the same properties as the trapezoidal rule. Both G- α and LTS are less accurate than MSSTC(3,4,5) in the low-frequency range.

Besides, with the same n , MSSTH(n) and MSSTC(n) are compared in Figs. 5, 6, 7, 8, 9, 10, 11, 12 and 13, where Figs. 5, 6 and 7 show the spectral radius (SR)

of the cases $n = 3, 4, 5$, respectively, Figs. 8, 9 and 10 show the percentage amplitude decay, Figs. 11, 12 and 13 show the percentage period elongation, all considering the undamped case. The generalized Padé approximation [14, 15], referred to as Padé(n), is also employed for comparison. It is known as the most accurate rational approximation of e^z by using

$$A(z) = \frac{(1 - \rho_\infty)P_{n-1,n}(z) + 2\rho_\infty P_{n,n}(z)}{(1 - \rho_\infty)Q_{n-1,n}(z) + 2\rho_\infty Q_{n,n}(z)} \quad (29)$$

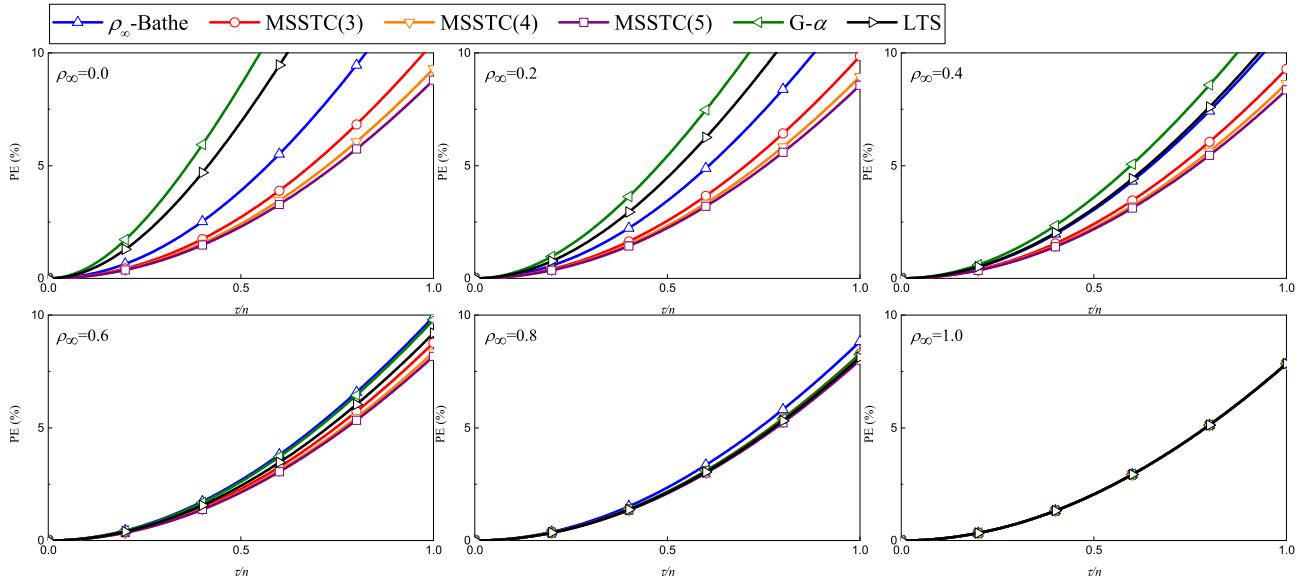


Fig. 4 Percentage period elongation for MSSTC(2,3,4,5), G- α and LTS

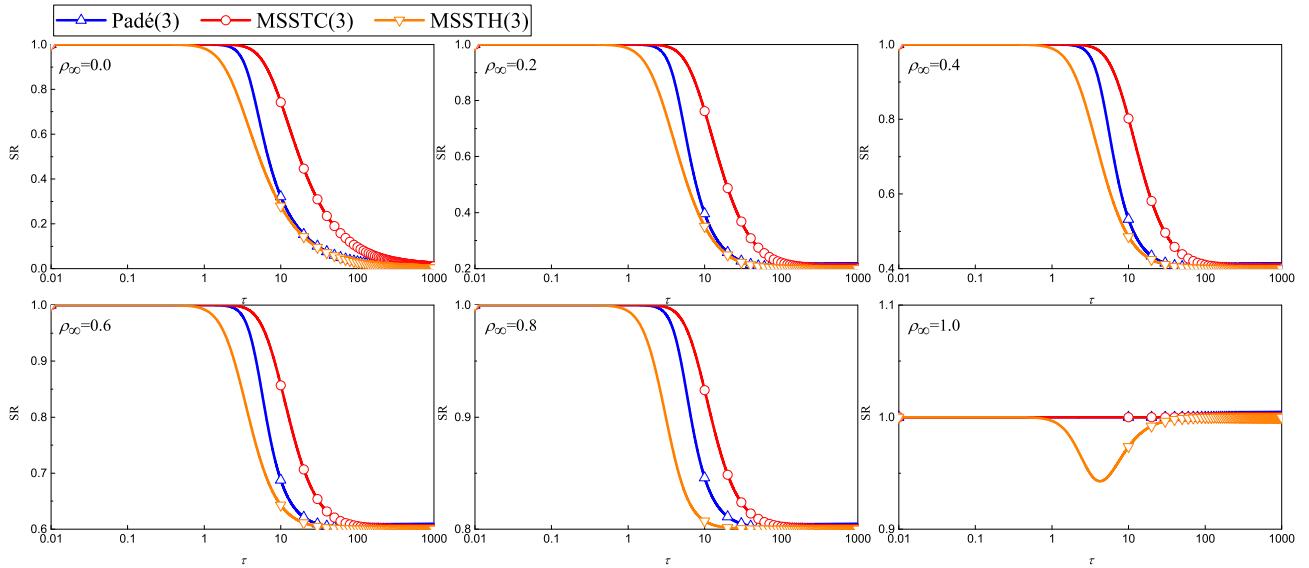


Fig. 5 Spectral radius for $n = 3$

where

$$P_{i,j}(z) = \sum_{p=0}^i \frac{i!(j+i-p)!}{(i-p)!(j+i)!} \frac{z^p}{p!} \quad (30a)$$

$$Q_{i,j}(z) = \sum_{p=0}^i (-1)^p \frac{j!(j+i-p)!}{(j-p)!(j+i)!} \frac{z^p}{p!} \quad (30b)$$

Padé(n) has $(2n - 1)$ th-order accuracy if $0 \leq \rho_\infty < 1$ and $(2n)$ th-order accuracy if $\rho_\infty = 1$.

As expected, Figs. 5, 6 and 7 demonstrate that MSSTC(n) preserves wider low-frequency range, followed by Padé(n), and MSSTH(n). Note that MSSTH(n) with $\rho_\infty = 1$ exhibits mild algorithmic dissipation in the medium frequency range, so these schemes are not recommended if all frequencies are requested. Figures 8, 9 and 10 also show that MSSTC(n) has the smallest amplitude dissipation in the low-frequency content. The amplitude decay ratio of MSSTC(5) is very close to 0 over $\tau \in [0, 2]$. In terms of period accuracy, Figs. 11, 12 and 13 show that Padé(n) is the most

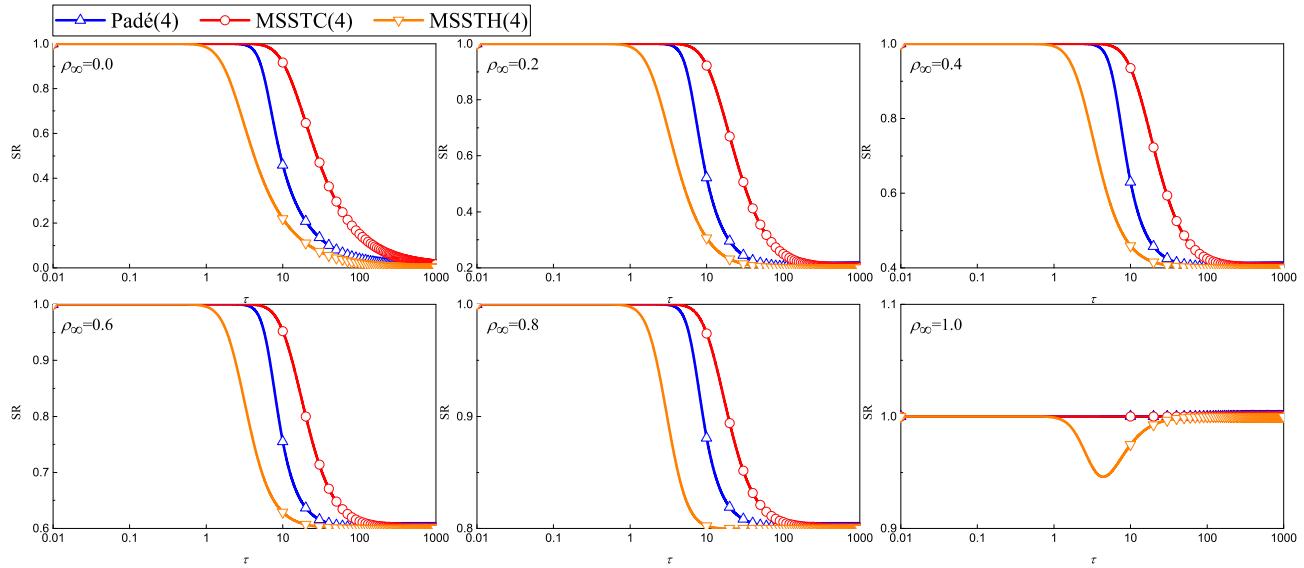


Fig. 6 Spectral radius for $n = 4$

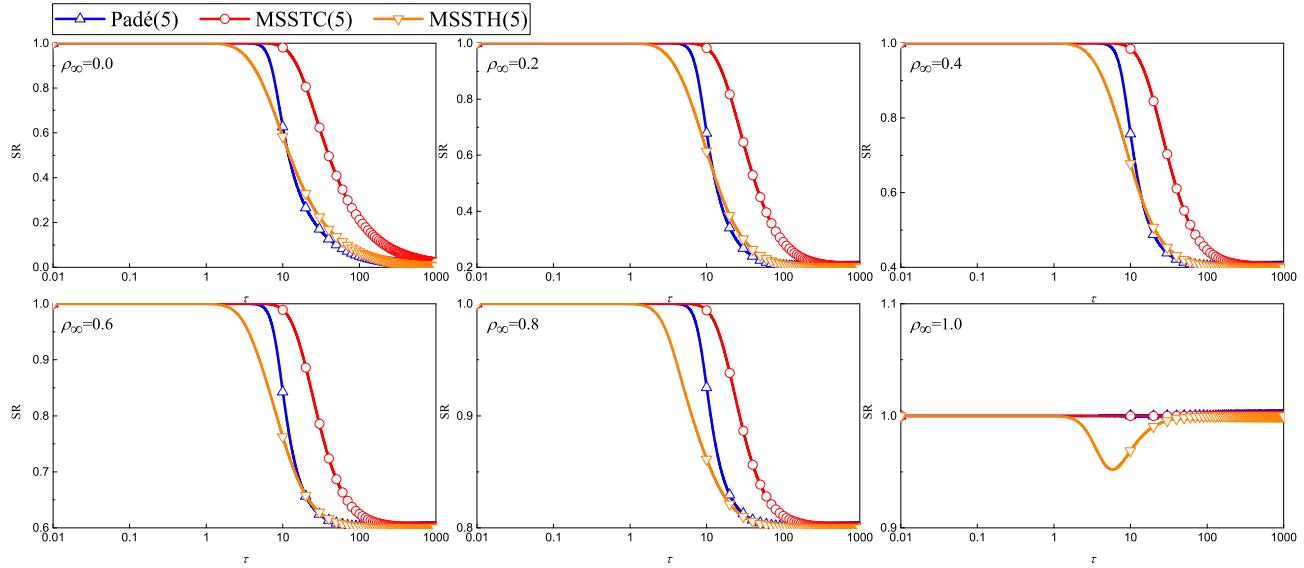


Fig. 7 Spectral radius for $n = 5$

accurate, followed by MSSTH(n), and MSSTC(n), consistent with the sequence of the accuracy order.

From the comparison, MSSTC(n) performs really good at conserving the low-frequency content, and its overall accuracy can be improved by using more substeps. MSSTH(n) shows higher period accuracy than the second-order methods, whereas its dissipation error is larger in the low-frequency content.

5 Numerical examples

To validate the performance, several numerical examples are solved in this section. As the spectral analysis has revealed the properties based on the linear model, this section focuses more on the application and discussion for nonlinear systems.

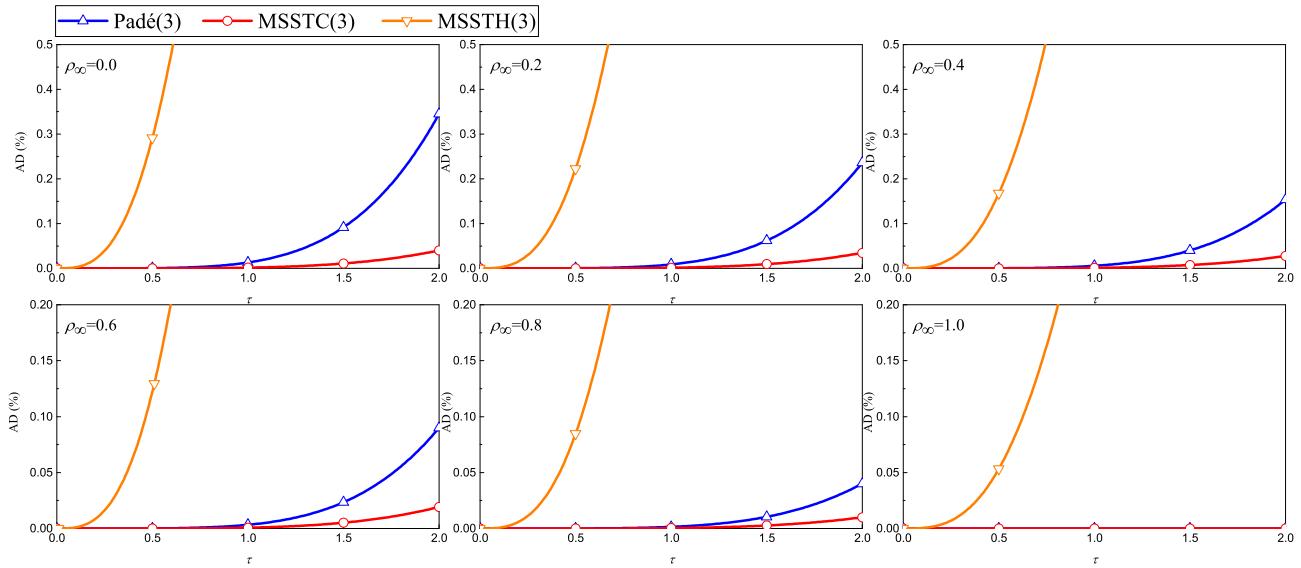


Fig. 8 Percentage amplitude decay for $n = 3$

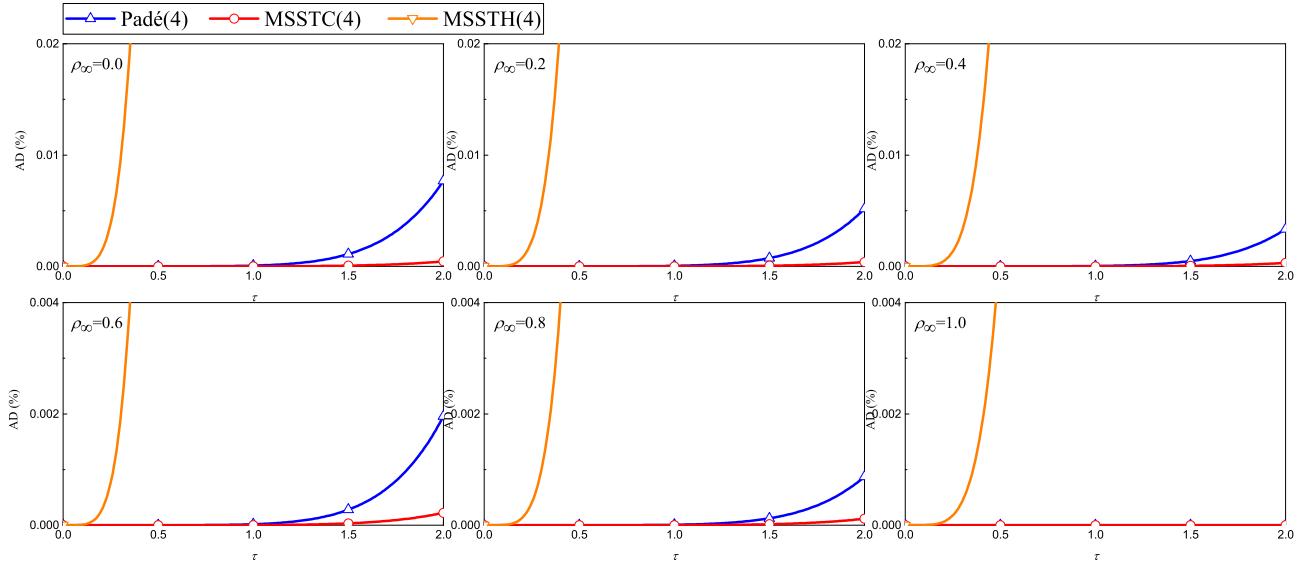


Fig. 9 Percentage amplitude decay for $n = 4$

5.1 Single degree-of-freedom examples

Firstly, two single degree-of-freedom examples, including a simple linear example and the nonlinear van der Pol's equation, are solved to check the convergence rate. The ρ_∞ -Bathe method, MSSTC(3,4,5) and MSSTH(3,4,5) with $\rho_\infty = 0.6$ is employed.

Linear example The linear equation of motion

$$\ddot{x} + 4x = 0, x(0) = 1, \dot{x}(0) = 1 \quad (31)$$

is considered, and the absolute errors of the displacement x_k , velocity \dot{x}_k , and acceleration \ddot{x}_k versus h at $t = 10$ are plotted in Fig. 14.

The results are consistent with the accuracy order. That is, MSSTC(n) and MSSTH(n) respectively present second-order and n th-order convergence rate. As a result, the higher-order MSSTH(n) enjoys significant accuracy advantage over the second-order methods. However, when h decreases from 10^{-2} , it seems that MSSTH(5) cannot maintain fifth-order accuracy. This is because when h is small enough, all effective num-

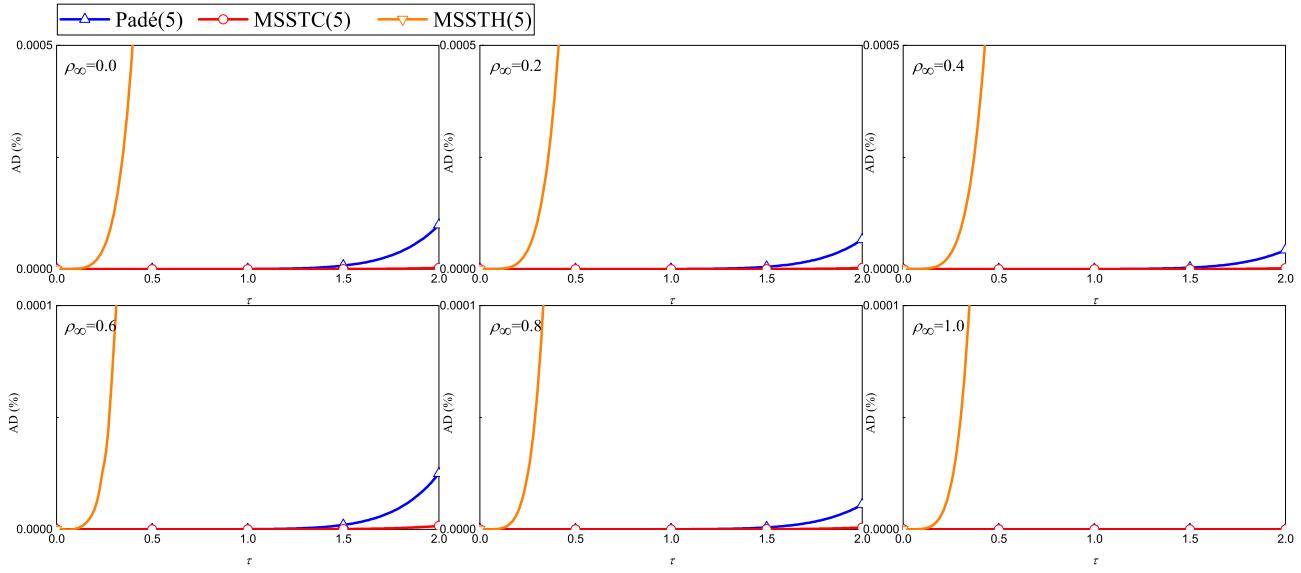


Fig. 10 Percentage amplitude decay for $n = 5$

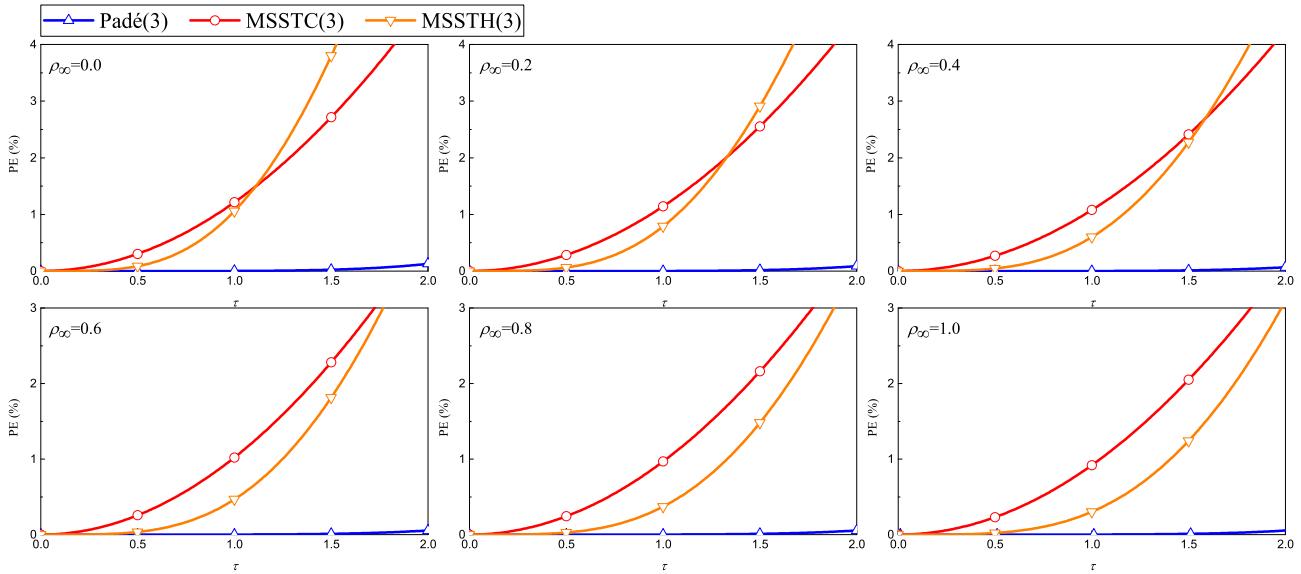


Fig. 11 Percentage period elongation for $n = 3$

bers stored in the computer are exactly precise, so if h continues to decrease, the accumulated rounding error can greatly spoil the numerical precision [31].

Van der Pol's equation The van der Pol's equation [19]

$$\begin{aligned} \dot{x}_1 &= x_2, \quad \dot{x}_2 = \epsilon^{-1}((1 - x_1^2)x_2 - x_1) \\ x_1(0) &= 2, \quad x_2(0) = -\frac{2}{3} + \frac{10}{81}\epsilon - \frac{292}{2187}\epsilon^2 + \frac{15266}{59049}\epsilon^3 \end{aligned} \quad (32)$$

is solved, where ϵ is an adjustable parameter. For the cases $\epsilon = 0.01, 0.001, 0.0001$, the absolute errors of $x_{1,k}$ and $x_{2,k}$ at $t = 1$ versus h are plotted in Fig. 15, where the reference solution is obtained by the ρ_∞ -Bathe method with $h = 10^{-7}$.

From Fig. 15, in most cases, the second- and n th-order convergence rate can be observed from errors of MSSTC(n) and MSSTH(n), respectively, but for the stiffer case of $\epsilon = 0.0001$, MSSTH(3) and MSSTH(5) show obvious order reduction in both x_1 and x_2 . It indi-

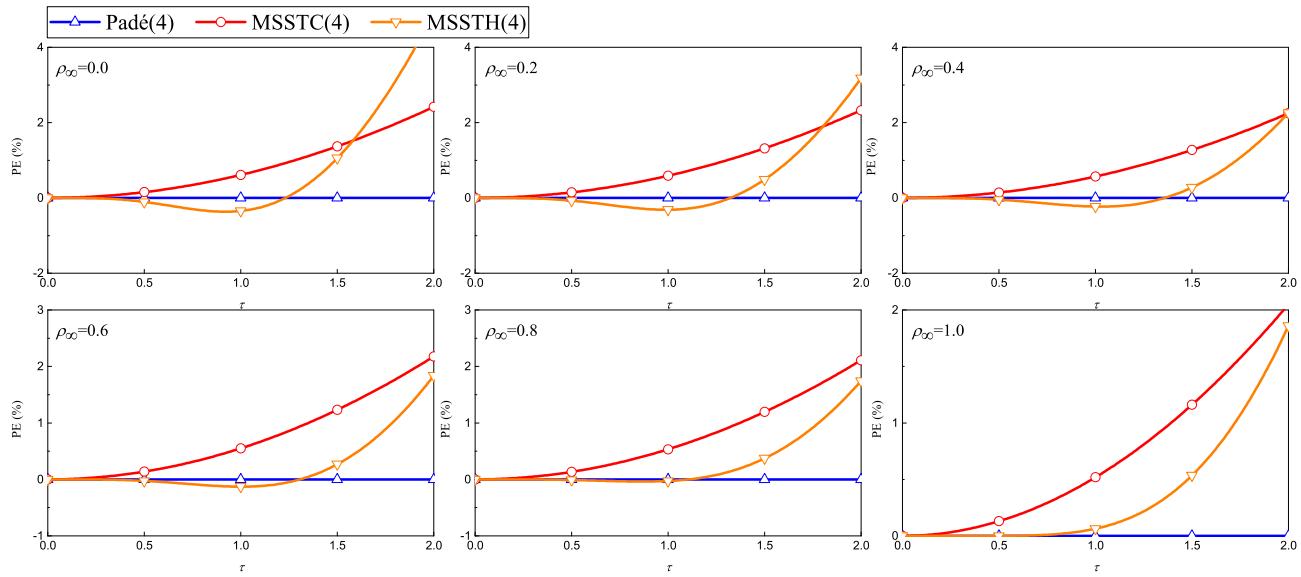


Fig. 12 Percentage period elongation for $n = 4$

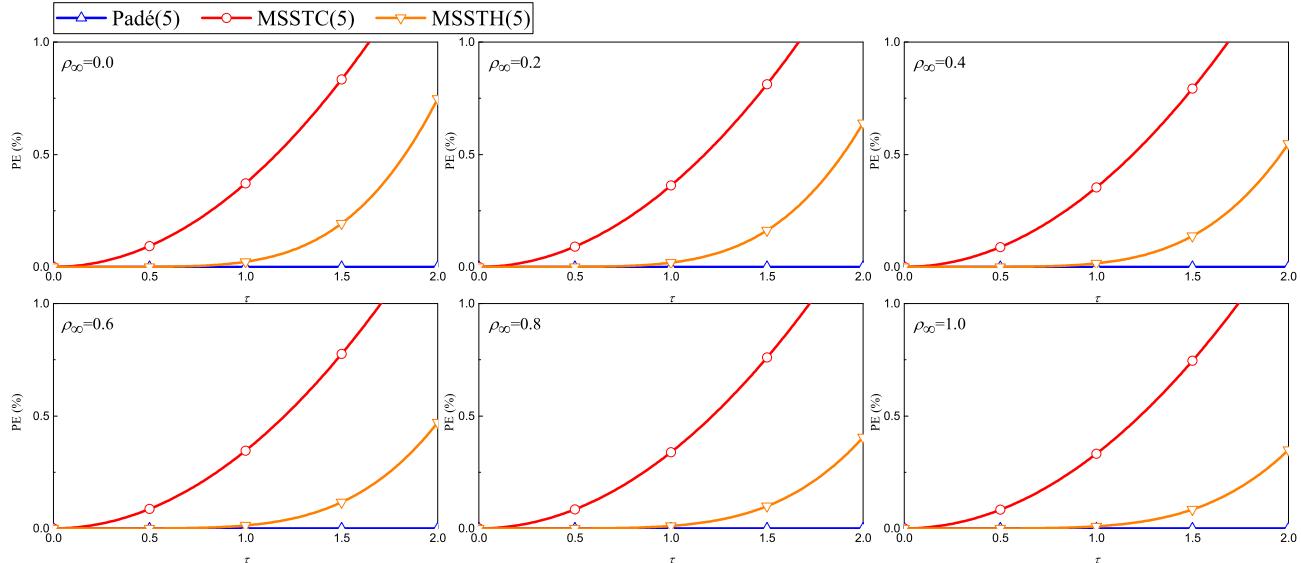


Fig. 13 Percentage period elongation for $n = 5$

cates that the accuracy order also depends on the problem to be solved when applied to nonlinear systems. The order reduction also occurs in other higher-order DIRKs when used for nonlinear problems, see Ref. [19]. Nevertheless, MSSTH(n) still shows significant accuracy advantage over the second-order MSSTC(n) with a small step size.

5.2 Multiple degrees-of-freedom examples

In this subsection, some illustrative examples are solved by using the ρ_∞ -Bathe method, MSSTC(3,4,5), MSSTH(3,4,5), G- α and LTS. In these methods, the parameter ρ_∞ is set as 0 uniformly, and the same h/n is used for comparison under close computational costs. The reference solutions are obtained by the ρ_∞ -Bathe method with an extremely small time step.

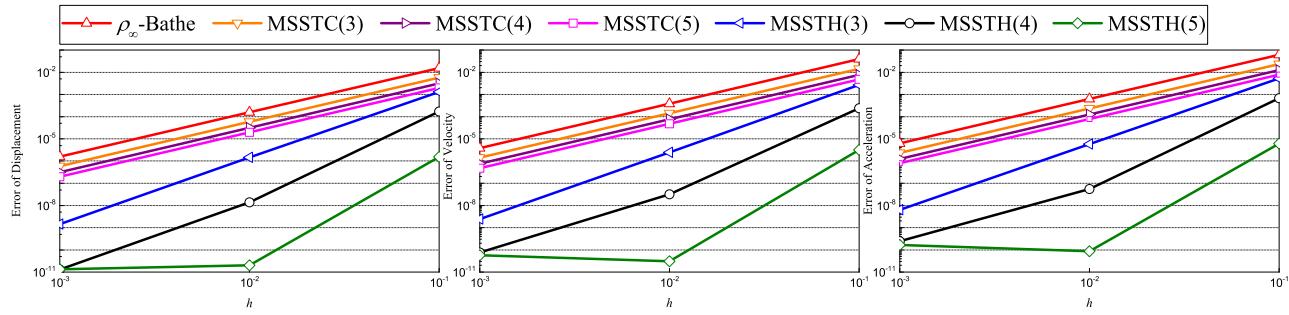


Fig. 14 Convergence rates for the single degree-of-freedom linear example

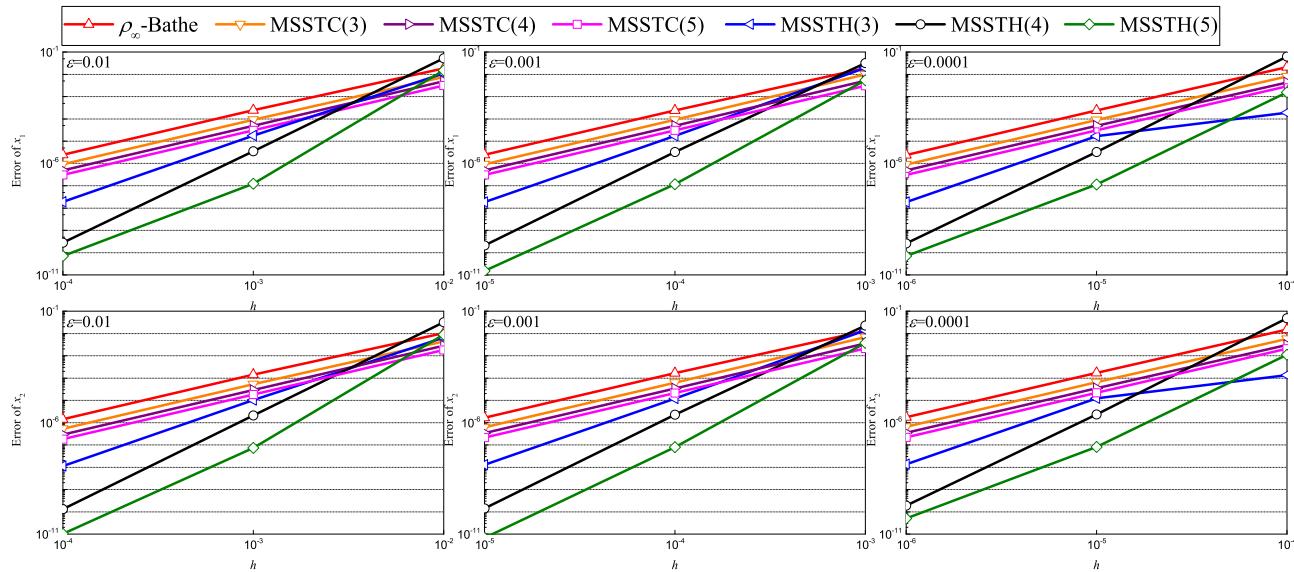


Fig. 15 Convergence rates for the van der Pol's equation

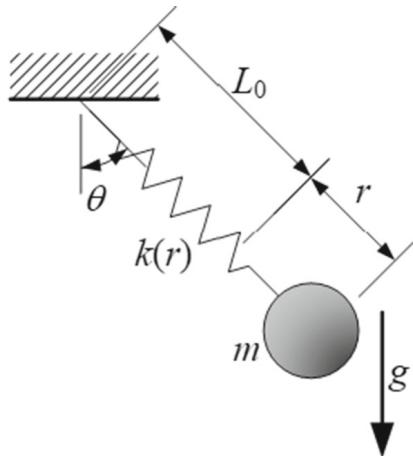


Fig. 16 Spring-pendulum model

Spring-pendulum model As shown in Fig. 16, the spring-pendulum model, where the spring is fixed at

one end and with a mass at the free end, is simulated. Its motion equation can be written as

$$m\ddot{r} + f(r) - m(L_0 + r)\dot{\theta}^2 - mg \cos \theta = 0 \quad (33a)$$

$$m\ddot{\theta} + \frac{m(2\dot{r}\dot{\theta} + g \sin \theta)}{L_0 + r} = 0 \quad (33b)$$

where $f(r)$ denotes the elastic force of the spring and other system parameters are assumed as $m = 1$ kg, $L_0 = 0.5$ m, $g = 9.81$ m/s². Three kinds of constitutive relations, as

$$f(r) = kr \quad (34a)$$

$$f(r) = kr^3 \quad (34b)$$

$$f(r) = k \tanh r \quad (34c)$$

Fig. 17 Numerical results of the spring-pendulum model ($f(r) = kr$, $k = 98.1 \text{ N/m}$)

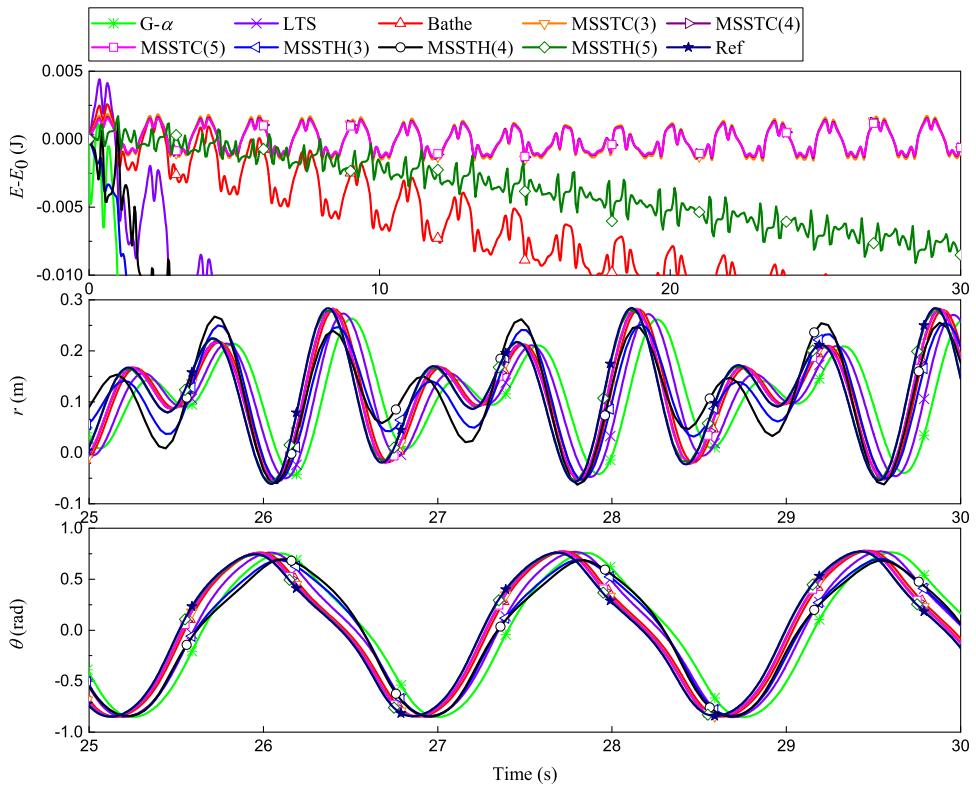


Fig. 18 Numerical results of the spring-pendulum model ($f(r) = kr^3$, $k = 98.1 \text{ N/m}$)

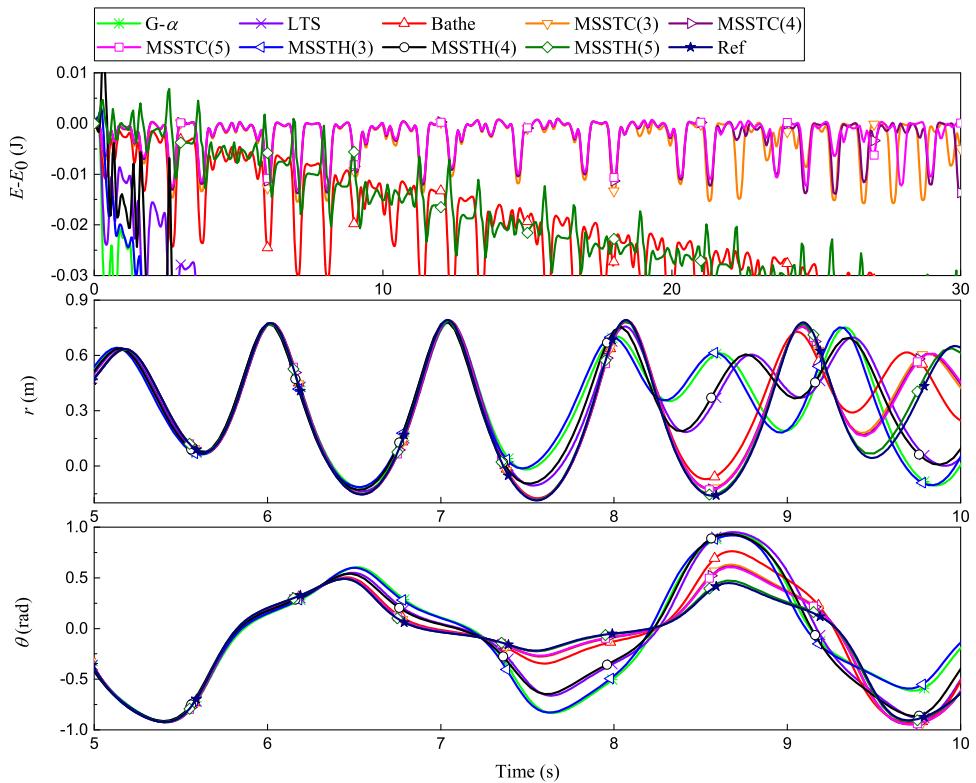
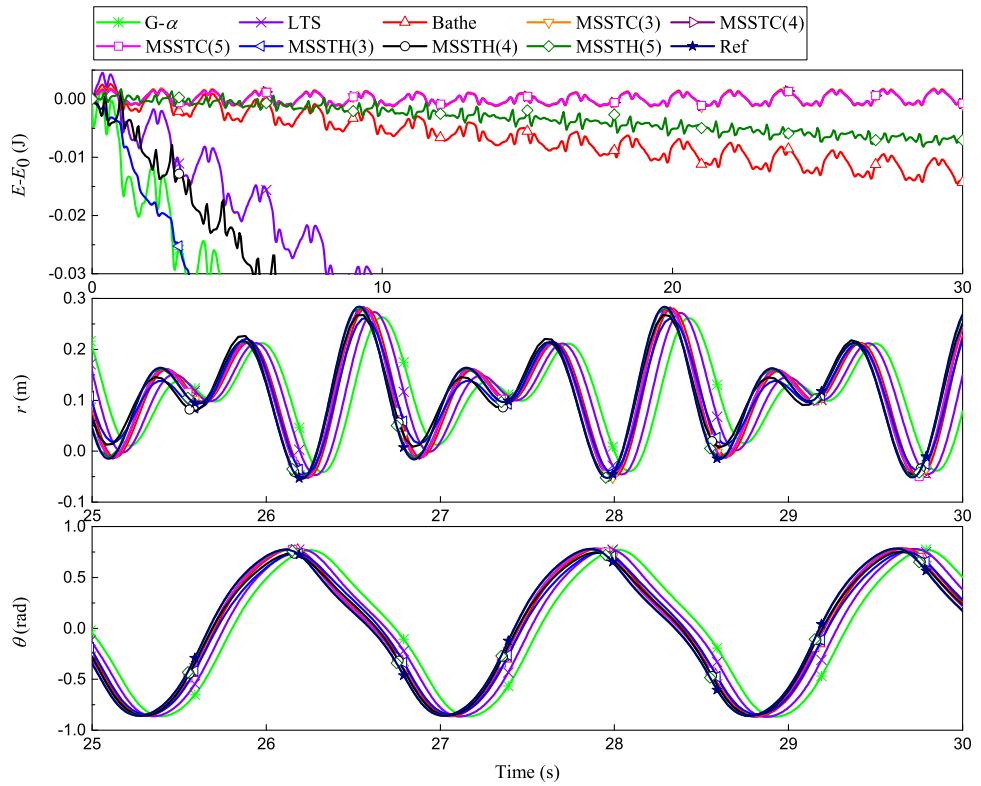


Fig. 19 Numerical results of the spring-pendulum model ($f(r) = k \tanh r$, $k = 98.1 \text{ N/m}$)



where $k = 98.1 \text{ N/m}$, are considered. The initial conditions are set as

$$r_0 = 0 \text{ m}, \dot{r}_0 = 1 \text{ m/s}, \theta_0 = \frac{\pi}{4} \text{ rad}, \dot{\theta}_0 \text{ rad/s} \quad (35)$$

Let $h/n = 0.01 \text{ s}$; the numerical solutions of $E - E_0$ (E denotes the system energy and E_0 is the initial value), r and θ for the three cases are summarized in Figs. 17, 18 and 19. From the curves of $E - E_0$, it can be observed that MSSTC(3,4,5) can almost preserve the numerical energy from decaying in all cases, despite the oscillations. MSSTH(5) can preserve more energy than the Bathe method, while G- α , LTS, and MSSTH(3,4) show obvious energy-decaying. From the numerical results of r and θ , one can see that with the step size, the numerical solutions of these methods have clearly deviated from the reference solution after a period of simulation. Among these methods, MSSTH(5) predicts the closest solutions to the reference ones, and G- α shows the largest errors. In addition, MSSTC(3,4,5) exhibit good amplitude accuracy thanks to their energy-preserving characteristic. These conclusions are all consistent with the results from linear analysis.

Moreover, to check the algorithmic dissipation, the stiff case, where $f(r) = kr$ ($k = 98.1 \times 10^{10} \text{ N/m}$), is also simulated with $h/n = 0.01 \text{ s}$. The numerical results of $E - E_0$, r and θ are plotted in Fig. 20. The results of r indicate that all employed schemes with $\rho_\infty = 0$ can effectively filter out the stiff component in the first few steps. After the initial decaying, MSSTC(3,4,5) can still preserve the remaining energy in the following simulation.

Slider-pendulum model The slider-pendulum model, shown in Fig. 21, is considered in this case. The slider is constrained by the spring, and one end of the pendulum is hinged to the center of mass of the slider. The motion is described by the differential-algebraic equations

$$m_1 \ddot{x}_1 + kx_1 = -\lambda_1 \quad (36a)$$

$$m_2 \ddot{x}_2 = \lambda_1 \quad (36b)$$

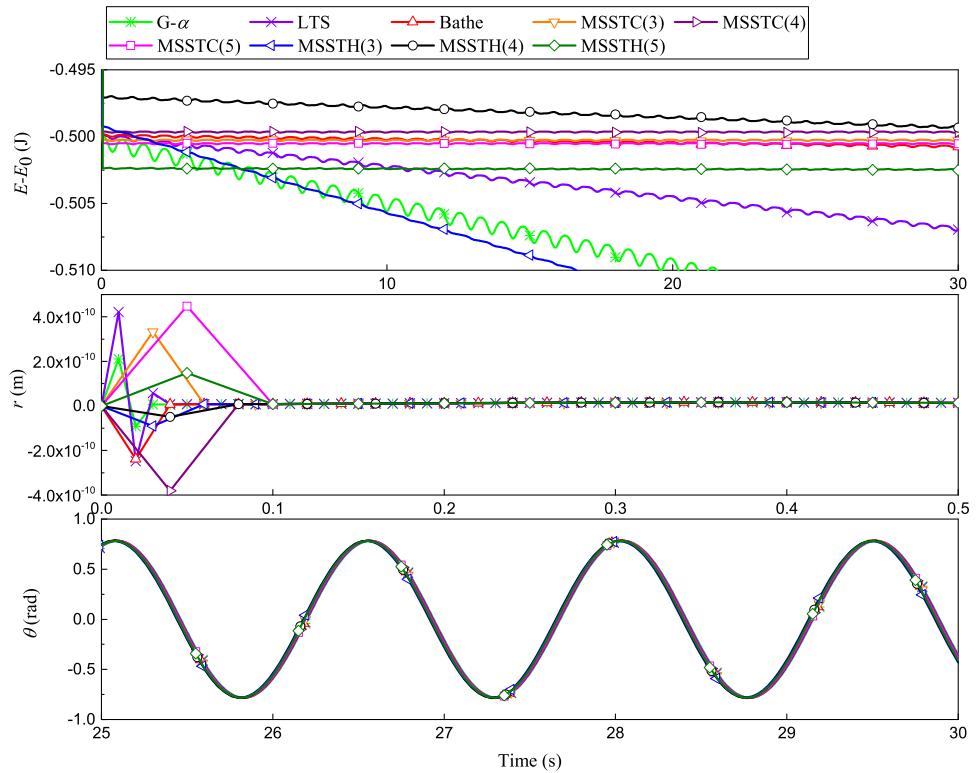
$$m_2 \ddot{y}_2 = \lambda_2 - m_2 g \quad (36c)$$

$$J_2 \ddot{\theta} = -\frac{L}{2} \lambda_1 \cos \theta - \frac{L}{2} \lambda_2 \sin \theta \quad (36d)$$

$$x_2 - x_1 = \frac{L}{2} \sin \theta \quad (36e)$$

$$y_2 = -\frac{L}{2} \cos \theta \quad (36f)$$

Fig. 20 Numerical results of the spring-pendulum model ($f(r) = kr$, $k = 98.1 \times 10^{10}$ N/m)



The system parameters are $m_1 = m_2 = 1$ kg, $L = 1$ m, $J_2 = \frac{1}{12}$ kg · m², $g = 9.81$ m/s², $k = 1$ N/m and 10^{10} N/m respectively for the compliant and stiff systems. The slider is excited by the initial horizontal velocity 1 m/s.

By using $h/n = 0.01$ s, the numerical solutions of $E - E_0$, x_1 and θ for the compliant and stiff cases are shown in Figs. 22 and 23, respectively. From the results of x_1 and θ , these methods all perform well in terms of accuracy and algorithmic dissipation. However, the numerical energies of MSSTH(4) show a slightly upward trend in the stiff case, so this method cannot give stable results for the problem.

As already discussed in several papers [5, 30], the unconditional stability of a time integration method derived from linear analysis cannot be guaranteed when they are applied to nonlinear problems. For nonlinear problems, the stability of a method depends not only on its recursive scheme, but also on the problem itself. Therefore, it is hard to give a definite conclusion about the stability of a method for general problems. From the numerical results, all employed methods, except MSSTH(4), provide stable results when solving stiff problems and differential-algebraic equations, so they can be said to have relatively strong stability.

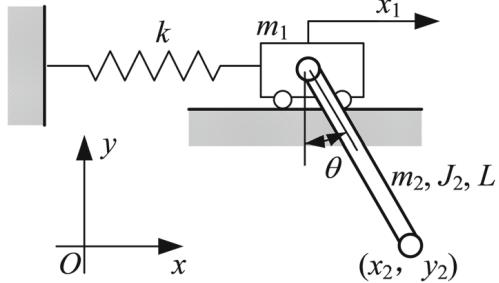


Fig. 21 Slider-pendulum model

MSSTH(4) is not recommended for these problems due to its poorer stability.

N-degree-of-freedom mass-spring system The N -degree-of-freedom mass-spring system [23], shown in Fig. 24, is considered to check the computational efficiency. The system parameters are set as

$$m_i = 1 \text{ kg}, f_i = \sin t \text{ N}, i = 1, 2, \dots, N \quad (37a)$$

$$k_i = \begin{cases} 10^5 \text{ N/m}, & i = 1 \\ 10^5 [1 - 2(x_i - x_{i-1})^2] \text{ N/m}, & 2 \leq i \leq N \end{cases} \quad (37b)$$

Fig. 22 Numerical results of the slider-pendulum model ($k = 1 \text{ N/m}$)

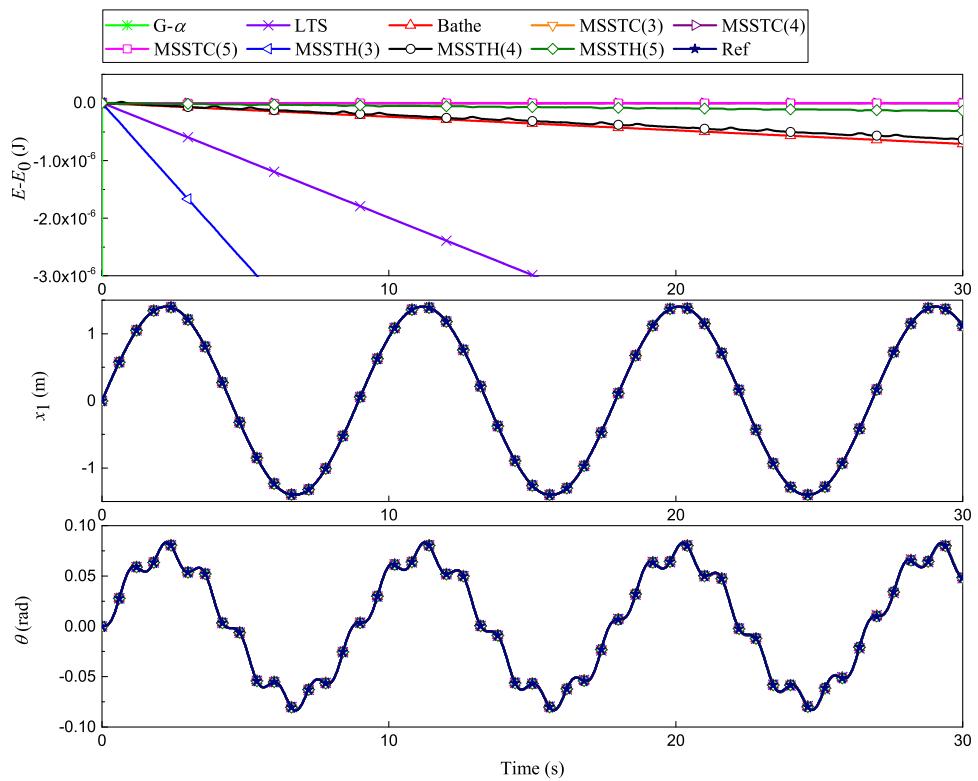
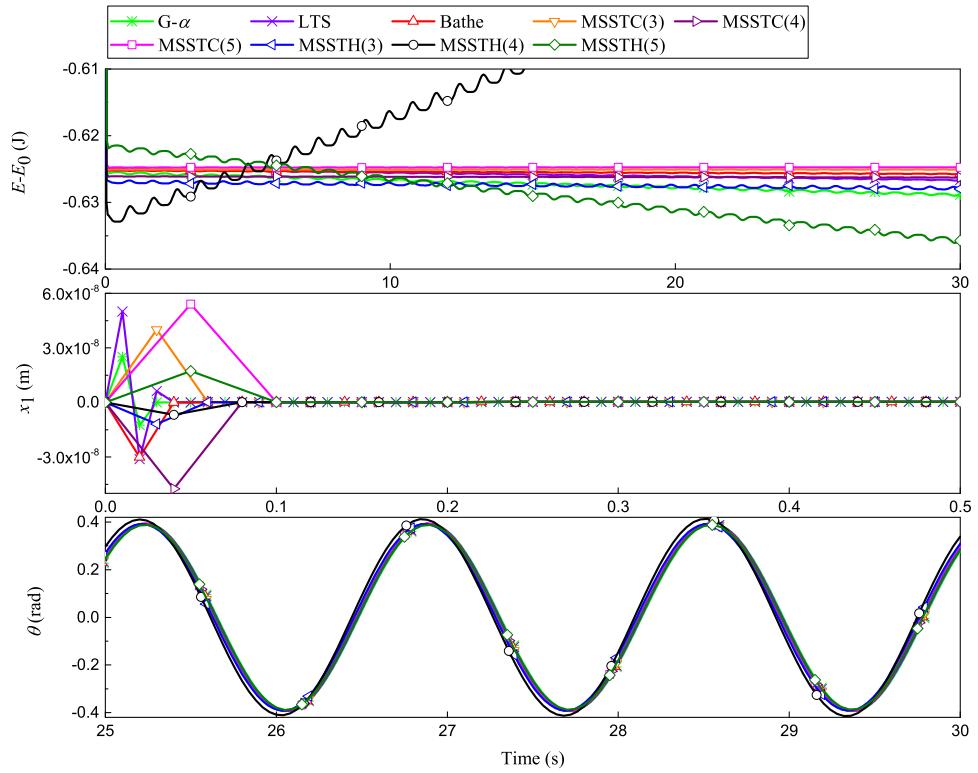


Fig. 23 Numerical results of the slider-pendulum model ($k = 10^{10} \text{ N/m}$)



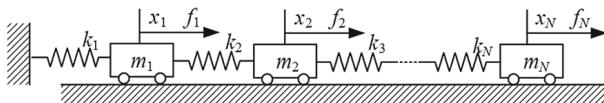


Fig. 24 Mass-spring model

With zero initial conditions, three cases, $N = 500, 1000$ and 1500 , are simulated by these methods using $h/n = 0.01$ s. Figure 25 shows the numerical solutions of x_N . It follows that with the step size, all methods can provide reliable results. The CPU time and total number of iterations required by these methods in the simulation of $[0, 30]$ s are summarized in Table 6. With $h/n = 0.01$ s, these methods need to proceed 3000 steps (sub-steps for the composite methods) in the whole simulation. Table 6 shows that in addition to MSSTH(4,5), other methods only require one iteration per step/sub-step, so their computational costs are almost equal to each other. One can also see that the required CPU time is approximately proportional to the number of iterations. MSSTH(4,5), especially MSSTH(4), take slightly longer time than other methods.

To check the generality of this conclusion, the required total number of iterations in the above spring-

pendulum and slider-pendulum examples are also listed in Table 7. In the two examples, $h/n = 0.01$ s is adopted, and the simulation of $[0, 30]$ s is also performed. The results indicate that the total numbers of iterations required by the second-order methods are very close in all cases. Although the higher-order methods need more iterations sometimes, the increased numbers, especially in MSSTH(3,5), are not very large in most cases. Therefore, it is reasonable to say that these methods with the same h/n have similar efficiency for nonlinear problems, and the above comparisons in terms of properties are conducted under the close computational costs.

Overall, the numerical examples in this section demonstrate that when applied to nonlinear problems, the proposed methods can still take advantage of their properties, including the energy-conserving characteristic of MSSTC(n), the high-accuracy of MSSTH(n), and the strong dissipation ability of both sub-families. However, MSSTH(n) shows reduced order and energy instability in some examples. From the presented solutions, MSSTH(5) is more recommended in the higher-order sub-family because of its high accuracy and robust stability, whereas MSSTH(4) is not so prefer-

Fig. 25 Computed x_N of the mass-spring model

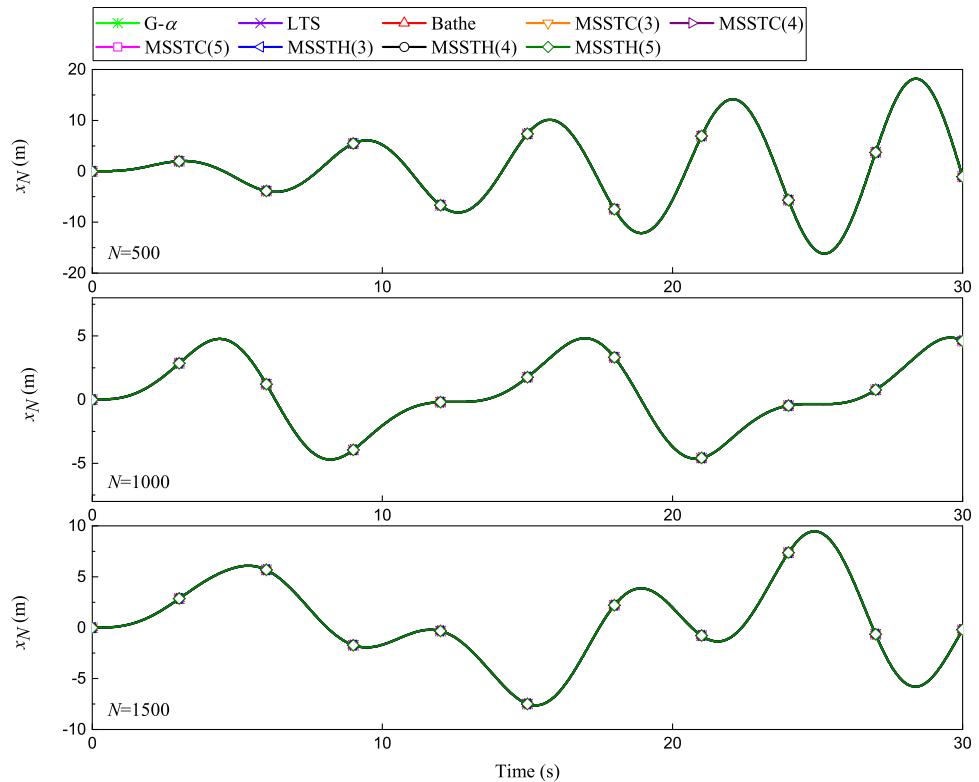


Table 6 CPU time and total number of iterations required by these methods in the mass-spring example

Method	$N = 500$		$N = 1000$		$N = 1500$	
	CPU time (s)	Number of iterations	CPU time (s)	Number of iterations	CPU time (s)	Number of iterations
G- α	34.8852	3000	155.0207	3000	364.3084	3000
LTS	34.1053	3000	164.0188	3000	361.6325	3000
Bathe	33.6191	3000	162.6447	3000	367.3689	3000
MSSTC(3)	34.2762	3000	164.4059	3000	363.7048	3000
MSSTC(4)	32.6436	3000	164.3387	3000	365.8208	3000
MSSTC(5)	33.3455	3000	162.9475	3000	361.8247	3000
MSSTH(3)	38.1994	3000	162.6028	3000	360.1350	3000
MSSTH(4)	52.6115	3988	165.4999	3012	383.2466	3306
MSSTH(5)	47.8183	3289	163.5020	3000	369.4852	3008

Table 7 Total number of iterations required by these methods in the spring-pendulum and slider-pendulum example

Method	Spring-pendulum example				Slider-pendulum example	
	$f(r) = kr$	$f(r) = kr^3$	$f(r) = k \tanh r$	$f(r) = kr$ (Stiff case)	Compliant case	Stiff case
G- α	5999	5999	5997	5925	3161	5980
LTS	5999	6000	5997	5923	3164	5978
Bathe	5998	5997	5997	5902	3285	6001
MSSTC(3)	5999	5998	5998	5890	3000	5978
MSSTC(4)	5999	5998	6000	5893	3000	6003
MSSTC(5)	5999	5999	5997	5891	3000	5992
MSSTH(3)	6000	6001	6000	5989	5751	6007
MSSTH(4)	8405	8286	8355	6006	5973	6663
MSSTH(5)	6704	6583	6676	5998	5828	6139

able, since it shows energy-instability and needs more iterations in some examples.

6 Conclusions

In this work, the n -sub-step composite method ($n \geq 2$), which employs the trapezoidal rule in the first $n - 1$ sub-steps and a general formula in the last one, is discussed. By optimizing the parameters, the two sub-families, named MSSTC(n) and MSSTH(n), are developed, respectively for the energy-conserving and high-accuracy purposes. From linear analysis, MSSTC(n) and MSSTH(n) are second-order and n th-order accurate, respectively, and they can both achieve unconditional stability with controllable algorithmic dissipation. In MSSTC(n), the purpose of energy-conserving is realized by maximizing the spectral radius in the low-frequency range.

A general approach of parameter optimization, suitable for all schemes with $n \geq 2$, is proposed; in this work, the cases $n = 2, 3, 4, 5$ are discussed in detail. When $n = 2$, both sub-families reduce to the ρ_∞ -Bathe method. As n increases, MSSTC(n) shows higher amplitude and period accuracy; its amplitude accuracy is even higher than that of the $(2n - 1)$ th-order Padé(n) approximation. MSSTH(3,4,5) exhibits lower period errors than the second-order methods, but their dissipation errors are larger.

The proposed methods are checked on several illustrative examples. The numerical results are mostly consistent with the conclusions from linear analysis. That is, MSSTC(n) can conserve the energy corresponding to the low-frequency content, and MSSTH(n) shows higher-order convergence rate for linear and nonlinear equations. However, in the nonlinear examples, some unexpected situations, such as order reduction

and energy instability, emerged in MSSTH(n). In this sub-family, MSSTH(5) is more recommended thanks to its high-accuracy and robust stability, and MSSTH(4) is not so preferable, since it shows energy instability and lower efficiency in these examples. However, these conclusions about nonlinear problems are obtained from the existing numerical results. The theoretical analysis is still desired in the future.

Acknowledgements The first and second authors acknowledge the financial support by the China Scholarship Council.

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bank, R.E., Coughran, W.M., Fichtner, W., Grosse, E.H., Rose, D.J., Smith, R.K.: Transient simulation of silicon devices and circuits. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **4**(4), 436–451 (1985)
- Bathe, K.J.: Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme. *Comput. Struct.* **85**(7–8), 437–445 (2007)
- Bathe, K.J., Baig, M.M.I.: On a composite implicit time integration procedure for nonlinear dynamics. *Comput. Struct.* **83**(31–32), 2513–2524 (2005)
- Bathe, K.J., Noh, G.: Insight into an implicit time integration scheme for structural dynamics. *Comput. Struct.* **98**, 1–6 (2012)
- Belytschko, T., Schoeberle, D.: On the unconditional stability of an implicit algorithm for nonlinear structural dynamics. *J. Appl. Mech.* (1975). <https://doi.org/10.1115/1.3423721>
- Butcher, J.C.: Implicit Runge–Kutta processes. *Math. Comput.* **18**(85), 50–64 (1964)
- Butcher, J.C.: Numerical Methods for Ordinary Differential Equations. John Wiley, New Jersey (2016)
- Chandra, Y., Zhou, Y., Stanciuescu, I., Eason, T., Spottswood, S.: A robust composite time integration scheme for snap-through problems. *Comput. Mech.* **55**(5), 1041–1056 (2015)
- Chung, J., Hulbert, G.: A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- α method. *J. Appl. Mech.* **60**, 371–375 (1993)
- Dahlquist, G.G.: A special stability problem for linear multistep methods. *BIT Numer. Math.* **3**(1), 27–43 (1963)
- Dong, S.: BDF-like methods for nonlinear dynamic analysis. *J. Comput. Phys.* **229**(8), 3019–3045 (2010)
- Fung, T.: Weighting parameters for unconditionally stable higher-order accurate time step integration algorithms. part 1—first-order equations. *Int. J. Numer. Methods Eng.* **45**(8), 941–970 (1999)
- Fung, T.: Weighting parameters for unconditionally stable higher-order accurate time step integration algorithms. Part 2—second-order equations. *Int. J. Numer. Methods Eng.* **45**(8), 971–1006 (1999)
- Fung, T.: Solving initial value problems by differential quadrature method. part 1: first-order equations. *Int. J. Numer. Methods Eng.* **50**(6), 1411–1427 (2001)
- Fung, T.: Solving initial value problems by differential quadrature method. part 2: second-and higher-order equations. *Int. J. Numer. Methods Eng.* **50**(6), 1429–1454 (2001)
- Gear, C.W.: Numerical Initial Value Problems in Ordinary Differential Equations. Prentice Hall PTR, New Jersey (1971)
- Hilber, H.M., Hughes, T.J., Taylor, R.L.: Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthq. Eng. Struct. Dyn.* **5**(3), 283–292 (1977)
- Ji, Y., Xing, Y.: An optimized three-sub-step composite time integration method with controllable numerical dissipation. *Comput. Struct.* **231**, 106210 (2020)
- Kennedy, C.A., Carpenter, M.H.: Diagonally implicit Runge–Kutta methods for stiff ODEs. *Appl. Numer. Math.* **146**, 221–244 (2019)
- Kim, W., Choi, S.Y.: An improved implicit time integration algorithm: the generalized composite time integration algorithm. *Comput. Struct.* **196**, 341–354 (2018)
- Kim, W., Reddy, J.: An improved time integration algorithm: a collocation time finite element approach. *Int. J. Struct. Stab. Dyn.* **17**(02), 1750024 (2017)
- Li, J., Yu, K., He, H.: A second-order accurate three sub-step composite algorithm for structural dynamics. *Appl. Math. Model.* **77**, 1391–1412 (2020)
- Li, J., Yu, K., Li, X.: A novel family of controllably dissipative composite integration algorithms for structural dynamic analysis. *Nonlinear Dyn.* **96**(4), 2475–2507 (2019)
- Masarati, P., Lanz, M., Mantegazza, P.: Multistep integration of ordinary, stiff and differential-algebraic problems for multibody dinamics applications. In: *Xvi Congresso Nazionale AIDAA*, pp. 1–10 (2001)
- Newmark, N.M.: A method of computation for structural dynamics. *J. Eng. Mech. Div.* **85**(3), 67–94 (1959)
- Noh, G., Bathe, K.J.: The Bathe time integration method with controllable spectral radius: the ρ_∞ -Bathe method. *Comput. Struct.* **212**, 299–310 (2019)

27. Noh, G., Ham, S., Bathe, K.J.: Performance of an implicit time integration scheme in the analysis of wave propagations. *Comput. Struct.* **123**, 93–105 (2013)
28. Tamma, K.K., Har, J., Zhou, X., Shimada, M., Hoitink, A.: An overview and recent advances in vector and scalar formalisms: space/time discretizations in computational dynamics—a unified approach. *Arch. Comput. Methods Eng.* **18**(2), 119–283 (2011)
29. Wood, W., Bossak, M., Zienkiewicz, O.: An alpha modification of Newmark's method. *Int. J. Numer. Methods Eng.* **15**(10), 1562–1566 (1980)
30. Xie, Y., Steven, G.P.: Instability, chaos, and growth and decay of energy of time-stepping schemes for non-linear dynamic equations. *Commun. Numer. Methods Eng.* **10**(5), 393–401 (1994)
31. Xing, Y., Zhang, H., Wang, Z.: Highly precise time integration method for linear structural dynamic analysis. *Int. J. Numer. Methods Eng.* **116**(8), 505–529 (2018)
32. Zhang, H., Xing, Y.: Optimization of a class of composite method for structural dynamics. *Comput. Struct.* **202**, 60–73 (2018)
33. Zhang, J.: A-stable two-step time integration methods with controllable numerical dissipation for structural dynamics. *Int. J. Numer. Methods Eng.* **121**, 54–92 (2020)
34. Zhou, X., Tamma, K.K.: Design, analysis, and synthesis of generalized single step single solve and optimal algorithms for structural dynamics. *Int. J. Numer. Methods Eng.* **59**(5), 597–668 (2004)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Appendix B

Performance of implicit A-stable time integration methods for multibody system dynamics

This appendix includes a verbatim copy of the paper “Performance of implicit A-stable time integration methods for multibody system dynamics”, written by Huimin Zhang, Runsen Zhang, Andrea Zanoni and Pierangelo Masarati and published in Multibody System Dynamics.

The paper gives the most recent description of MBDyn’s different integrators properties.

The paper is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <https://creativecommons.org/licenses/by/4.0/>



Performance of implicit A-stable time integration methods for multibody system dynamics

Huimin Zhang^{1,2} · Runsen Zhang^{1,2} · Andrea Zanoni² · Pierangelo Masarati²

Received: 30 June 2021 / Accepted: 22 October 2021 / Published online: 18 January 2022
© The Author(s) 2022

Abstract

This paper illustrates the performance of several representative implicit A-stable time integration methods with algorithmic dissipation for multibody system dynamics, formulated as a set of mixed implicit first-order differential and algebraic equations. The integrators include the linear multi-step methods with two to four steps, the single-step reformulations of the linear multi-step methods, and explicit first-stage, singly diagonally-implicit Runge–Kutta methods. All methods are implemented in the free, general-purpose multibody solver MBDyn. Their formulations and implementation are presented. According to the comparison from linear analysis and numerical experiments, some general conclusions on the selection of integration schemes and their implementation are obtained. Although all of these methods can predict reasonably accurate solutions, the specific advantages that each of them has in different situations are discussed.

Keywords Implicit · A-stability · Time integration methods · Multibody system dynamics

1 Introduction

Multibody system dynamics problems can be typically formulated as a set of Differential-Algebraic Equations (DAEs), often in semi-explicit form. The numerical treatment of DAEs is more challenging than that of Ordinary Differential Equations (ODEs). Typically, two strategies are employed: direct discretization of DAEs and discretization after reformulation [3]. Reformulation usually consists of some sort of index reduction that can convert DAEs into ODEs, and thus allows the problems to be solved using relatively conventional methods. However, this process may be costly, since it may require substantial user intervention and may be convenient only when a substantial reduction in coordinates can be achieved, which is not the case for examples where mechanisms are analyzed made of flexible components, so direct discretization has gained more attention in software development.

Direct discretization poses strict requirements on time integration methods. Explicit integrators, such as the central difference method [13] and the explicit Runge–Kutta methods [10], are difficult to use to solve DAEs directly because they cannot satisfy the algebraic

✉ H. Zhang
huimin.zhang@polimi.it

¹ School of Aeronautic Science and Engineering, Beihang University, Beijing 100083, China

² Dipartimento di Scienze e Tecnologie Aeroespaziali, Politecnico di Milano, Milano 20156, Italy

constraint equations at the position level. Implicit integrators, including the linear multi-step methods [23, 24], the implicit Runge–Kutta methods [9] and many direct integration methods [30], are required. They are designed to possess good accuracy, A-stability (unconditional stability), and tunable algorithmic dissipation to provide accurate and robust solutions.

Implicit time integrators are briefly reviewed here. In the field of structural dynamics, several single-step single-solve methods, such as the Newmark method [25], the Wilson- θ method [31], the HHT- α method [17] (proposed by Hilber, Hughes and Taylor), the generalized- α method [11, 22], the GSSSS (Generalized Single-Step Single-Solve) method [37], and many others [18, 32], have been developed since the 1950s. Most of these methods have second-order accuracy, A-stability and tunable algorithmic dissipation from linear analysis. They were initially designed to solve second-order ODEs in structural dynamics, and some of their improved formulations can also be used to solve DAEs and general first-order differential equations [2, 8, 20]. Some comparisons between the linear two-step method and single-step methods have already been presented in [34, 36] and are not reproduced here. Therefore, these single-step methods are not considered further in this work.

In the class of multi-step methods, the linear two-step method [24], and several backward difference formulas (BDFs) [19, 28], have been efficiently used in multibody system dynamics. The optimal parameters of the linear three- and four-step methods with second-order accuracy, A-stability and tunable algorithmic dissipation were given in [36]. According to Dahlquist's second barrier [12], the linear multi-step methods cannot exceed second-order accuracy to possess A-stability. To eliminate the additional starting procedures of the multi-step methods, their equivalent single-step reformulations, obtained by introducing a few auxiliary variables, have been also proposed in [36]. These multi-step and equivalent single-step integrators are designed for systems of first-order differential equations, and can be generalized to solve second- and higher-order differential equations in a straightforward way.

Another important branch of implicit integrators are the multi-stage methods, represented by the Runge–Kutta family [10]. They evaluate the states at intermediate time points per step, and compute the states at discrete time points using a scheme like the quadrature formula. For solving DAEs, the stiffly-accurate Runge–Kutta methods without the quadrature step are more practical, because the constraints are satisfied at the final stage, but may not be satisfied after implementing the quadrature formula. Multi-stage methods can be designed to have higher-order accuracy and A-stability simultaneously, as in [1, 6, 21, 35]. Considering the computational cost, the singly diagonally-implicit Runge–Kutta methods [27], which perform the computation of each stage in sequence, are more convenient and recommended. Consequently, a few recently proposed second- and higher-order stiffly-accurate, singly diagonally-implicit Runge–Kutta methods with explicit first-stage [21, 35] are employed in this work.

The purpose of this work is to present a comparative study of several representative implicit, A-stable and algorithmically dissipative time integration methods for multibody system dynamics. The time integration methods employed include the linear multi-step methods [23, 24, 36], their equivalent single-step methods [36], and stiffly-accurate, explicit first-stage, singly diagonally-implicit Runge–Kutta (ESDIRK) methods [21, 35]. These methods are implemented in the free general-purpose multibody dynamics analysis software MB-Dyn¹ [24]. Their formulation is presented in Sect. 2, whereas Sect. 3 describes their implementation in MB-Dyn. Considering a linear problem, the accuracy, algorithmic dissipation

¹<https://www.mbdyn.org>.

and dispersion properties of these methods are discussed in Sect. 4. These methods are then applied to solve of some benchmark multibody dynamics problems in Sect. 5. By comparing the numerical results, several general conclusions on the implementation and selection of time integration methods are finally summarized in Sect. 6.

2 Formulation

Initial-value problems in MBDyn are formulated as a set of implicit first-order DAEs, whose general form is

$$\mathbf{r}(\mathbf{y}, \dot{\mathbf{y}}, t) = \mathbf{0}, \quad \mathbf{y}(t_0) = \mathbf{y}_0 \quad (1)$$

where \mathbf{y} collects the differential and algebraic variables, and the overdot indicates the derivative with respect to time t . The initial condition \mathbf{y}_0 is given, and $\dot{\mathbf{y}}_0$ needs to be solved from $\mathbf{r}(\mathbf{y}_0, \dot{\mathbf{y}}_0, t_0) = \mathbf{0}$. For a detailed problem description, the reader should refer to [24].

Problems associated with constrained dynamics, as discussed later in Sect. 5, are formulated as Index-3 DAEs, by directly enforcing the holonomic kinematic constraints on the position level through algebraic relations between the coordinates of the problem using Lagrange multipliers. No elaborated scaling strategy is used, except for scaling the Jacobian matrix contribution by the inverse of $\Delta t b_0$, the coefficient that multiplies the derivative of the state at the current time step in implicit numerical schemes, as proposed in [7]. No hidden constraints, e.g., on the velocity level, are considered.

2.1 Linear multi-step methods

The linear multi-step methods can be expressed as

$$\mathbf{y}_k = \sum_{j=1}^r \alpha_j \mathbf{y}_{k-j} + \Delta t \sum_{j=0}^r \beta_j \dot{\mathbf{y}}_{k-j} \quad (2)$$

where \mathbf{y}_k and $\dot{\mathbf{y}}_k$ represent the numerical solution at time t_k , $\Delta t = t_k - t_{k-1}$ is the time step size, α_j and β_j are control parameters of the method. Considering the conditions of second-order accuracy, A-stability, and tunable algorithmic dissipation, the optimal parameters of the linear two-, three-, and four-step methods, referred to as LMS2, LMS3 and LMS4, can be found in [36]. They are controlled by a single parameter $\rho_\infty \in [0, 1]$, to adjust the degree of algorithmic dissipation. The algorithmic dissipation becomes stronger as ρ_∞ decreases. When $\rho_\infty = 0$, they reduce to a second-order BDF. In [24], the parameters of LMS2 used in variable-time-step cases were given, while so far LMS3 and LMS4 were previously formulated only for fixed time steps. The initialization of the integration procedure of the multi-step methods, is performed employing the trapezoidal rule for the first few steps

$$\mathbf{y}_k = \mathbf{y}_{k-1} + \frac{\Delta t}{2} (\dot{\mathbf{y}}_k + \dot{\mathbf{y}}_{k-1}) \quad (3)$$

which is A-stable and second-order accurate.

2.2 Equivalent single-step methods

The single-step method proposed in [36], which can be used as an equivalent alternative of the linear multi-step method, has the form

$$\mathbf{y}_k = \mathbf{y}_{k-1} + \Delta t((1 - \gamma_0)\dot{\mathbf{y}}_{k-1}^{r-1} + \gamma_0\dot{\mathbf{y}}_k^{r-1}) \quad (4a)$$

$$(1 - \gamma_1)\dot{\mathbf{y}}_{k-1}^{r-1} + \gamma_1\dot{\mathbf{y}}_k^{r-1} = (1 - \gamma_2)\dot{\mathbf{y}}_{k-1}^{r-2} + \gamma_2\dot{\mathbf{y}}_k^{r-2} \quad (4b)$$

$$(1 - \gamma_3)\dot{\mathbf{y}}_{k-1}^{r-2} + \gamma_3\dot{\mathbf{y}}_k^{r-2} = (1 - \gamma_4)\dot{\mathbf{y}}_{k-1}^{r-3} + \gamma_4\dot{\mathbf{y}}_k^{r-3} \quad (4c)$$

$$\dots \quad (4d)$$

$$(1 - \gamma_{2r-3})\dot{\mathbf{y}}_{k-1}^1 + \gamma_{2r-3}\dot{\mathbf{y}}_k^1 = (1 - \gamma_{2r-2})\dot{\mathbf{y}}_{k-1} + \gamma_{2r-2}\dot{\mathbf{y}}_k \quad (4e)$$

where $\dot{\mathbf{y}}_k^j (j = 1, 2, \dots, r-1)$ are auxiliary variables, and γ_i are control parameters. At the beginning, $\dot{\mathbf{y}}_0^1 = \dot{\mathbf{y}}_0^2 = \dots = \dot{\mathbf{y}}_k^{r-1} = \dot{\mathbf{y}}_0$ is used. The parameters of the single-step methods equivalent to LMS2, LMS3 and LMS4, referred to as SS2, SS3 and SS4, respectively, were also given in [36]. SS3 and SS4 have complex parameters, so the auxiliary variables they produce may be complex numbers, but the state variables are real, as demonstrated in [36].

2.3 Explicit first-stage, singly diagonally-implicit Runge–Kutta (ESDIRK) methods

The stiffly-accurate s -stage ESDIRK is represented using Butcher's tableau [10], as

0	0	0	0	...	0	0
c_2	a_{21}	γ	0	...	0	0
c_3	a_{31}	a_{32}	γ	..	0	0
\vdots	\vdots	\vdots	\vdots	\vdots
c_{s-1}	$a_{s-1,1}$	$a_{s-1,2}$	$a_{s-1,3}$...	γ	0
1	b_1	b_2	b_3	...	b_{s-1}	γ
	b_1	b_2	b_3	...	b_{s-1}	γ

where γ, a_{ij}, c_i, b_i are control parameters. Considering the implicit first-order equation (1), the intermediate stages are formulated as

$$\mathbf{y}_{k-1+c_i} = \mathbf{y}_{k-1} + \Delta t \left(\sum_{j=1}^{i-1} a_{ij} \dot{\mathbf{y}}_{k-1+c_j} + \gamma \dot{\mathbf{y}}_{k-1+c_i} \right) \quad (5)$$

$$i = 2, 3, 4, \dots, s-1$$

The last stage is

$$\mathbf{y}_k = \mathbf{y}_{k-1} + \Delta t \left(\sum_{j=1}^{s-1} b_j \dot{\mathbf{y}}_{k-1+c_j} + \gamma \dot{\mathbf{y}}_k \right) \quad (6)$$

where $c_1 = 0$ and $c_s = 1$. The computation of each stage can be implemented in sequence. Except for the explicit first stage, every other stage uses an implicit single-step or multi-step formula. They share the same effective stiffness matrix for linear analysis and the same form

of Jacobian matrix in nonlinear iterations. In the procedure, the state variables at the time points are given, but those at the intermediate stages are not output. Therefore, the multi-stage methods are essentially single-step schemes, which only use the information from the last step to update the current one.

The two-sub-step ρ_∞ -Bathe method [26], which can be seen as a 3-stage ESDIRK with second-order accuracy, A-stability and tunable algorithmic dissipation, is presented. Its parameters are

$$\gamma = \begin{cases} \frac{2 - \sqrt{2(1 + \rho_\infty)}}{2(1 - \rho_\infty)}, & \rho_\infty \in [0, 1) \\ \frac{1}{4}, & \rho_\infty = 1 \end{cases} \quad (7)$$

$$b_1 = -\frac{4\gamma^2 - 6\gamma + 1}{4\gamma}, \quad b_2 = \frac{1 - 2\gamma}{4\gamma}$$

The multi-sub-step methods MSSTC(n) and MSSTH(n), as $(n + 1)$ -stage ESDIRKs proposed in [35], are also employed. MSSTC(n) is designed to have second-order accuracy, A-stability, tunable algorithmic dissipation, preserving low-frequency dynamics as much as possible. It employs the trapezoidal rule from the second to the n th stage, and a general formula in the last one. The optimal parameters of MSSTC(3, 4, 5) were given in [35].

MSSTH(n) is designed to have n th-order accuracy, A-stability, and tunable algorithmic dissipation. However, since only linear analysis was considered in [35], its parameters are modified here to satisfy the overall-order and stage-order conditions for Runge–Kutta methods [10, 16, 21], without changing the linear characteristics. The design of the modified MSSTH(3, 4, 5) considers the following conditions:

- Each stage, from the second-one, has at least second-order accuracy;
- The overall accuracy is n th-order, and on this basis, the local truncation error is minimized;
- A-stability and tunable algorithmic dissipation for linear analysis.

Under the premise of A-stability, MSSTH(n) is designed to achieve the highest possible accuracy for a given ρ_∞ . The details of the parameter selection are shown in the [Appendix](#).

Besides, several stiffly-accurate ESDIRKs developed in [21], including the third-order 4-stage ESDIRK3(2)4L[2]SA, the third-order 5-stage ESDIRK3(2)5L[2]SA, and the fourth-order 6-stage ESDIRK4(3)6L[2]SA₂, are also employed. These methods are designed to be L-stable ($\rho_\infty = 0$), and their internal stages are also designed to be L-stable whenever possible. Their parameters can be found in [21].

3 Implementation

The detailed problem description, solution phases and implementation structure of MB-Dyn were presented in [24]. Time integrators are defined in the class `ImplicitStepIntegrator`. The multi-step, single-step, and multi-stage integrators are constructed using the template classes `tplStepNIntegrator`, `tplSingleStepIntegrator` and `tplStageNIntegrator`, respectively. They provide an `Advance()` method to perform a complete step. For multi-stage integrators, the operations of all stages are encapsulated in the method `Advance()`, so the solutions in the internal stages are hidden.

With the initial condition at t_0 , the multi-step and single-step integrators sequentially calculate the state variables at $t_1, t_2, t_3, \dots, t_k, \dots$, and the ESDIRKs need to solve the results at $t_{0+c_2}, t_{0+c_3}, \dots, t_{0+c_{s-1}}, t_1, t_{1+c_2}, t_{1+c_3}, \dots, t_{1+c_{s-1}}, t_2, \dots, t_k, \dots$ in turn. Note that the explicit first stage of ESDIRKs does not require any calculation. All methods show a very similar structure at each time point; the difference lies in the parameters and the number of previous time points used. The single-step methods, must additionally solve and store the auxiliary variables. Therefore, the implementation at a certain discrete time point, t_N , which is used to represent all time points, including those at the end of each step t_k and the internal time points of the ESDIRKs, is explained uniformly in this section. For multi-step and single-step methods, the information at t_{N-1}, t_{N-2}, \dots , used at the current time point t_N are the states of previous steps. For ESDIRKs, the information of t_{N-1}, t_{N-2}, \dots , are the states of the previously solved time points, which can be the last step and last stages.

At the discrete time point t_N , the state variables \mathbf{y}_N and $\dot{\mathbf{y}}_N$ are obtained by solving

$$\mathbf{r}(\mathbf{y}_N, \dot{\mathbf{y}}_N, t_N) = \mathbf{0} \quad (8a)$$

$$\mathbf{y}_N = \mathbf{f}(\mathbf{y}_{N-1}, \mathbf{y}_{N-2}, \dots, \dot{\mathbf{y}}_N, \dot{\mathbf{y}}_{N-1}, \dot{\mathbf{y}}_{N-2}, \dots, \dot{\mathbf{y}}_{N-1}^1, \dot{\mathbf{y}}_{N-1}^2, \dots, \dot{\mathbf{y}}_{N-1}^r) \quad (8b)$$

Here Eq. (8b) represents the integration scheme. It is Eq. (2) for multi-step integrators, and Eq. (5) or Eq. (6) for the ESDIRKs. For single-step methods, it needs to be reorganized from Eqs. (4a)–(4e). SS2 uses

$$\mathbf{y}_N = \mathbf{y}_{N-1} + \Delta t \left(\frac{\gamma_0 \gamma_2}{\gamma_1} \dot{\mathbf{y}}_N + \frac{\gamma_0(1 - \gamma_2)}{\gamma_1} \dot{\mathbf{y}}_{N-1} + \frac{\gamma_1 - \gamma_0}{\gamma_1} \dot{\mathbf{y}}_{N-1}^1 \right) \quad (9)$$

SS3 uses

$$\begin{aligned} \mathbf{y}_N &= \mathbf{y}_{N-1} \\ &+ \Delta t \left(\frac{\gamma_0 \gamma_2 \gamma_4}{\gamma_1 \gamma_3} \dot{\mathbf{y}}_N + \frac{\gamma_0 \gamma_2(1 - \gamma_4)}{\gamma_1 \gamma_3} \dot{\mathbf{y}}_{N-1} + \frac{\gamma_0(\gamma_3 - \gamma_2)}{\gamma_1 \gamma_3} \dot{\mathbf{y}}_{N-1}^1 + \frac{\gamma_1 - \gamma_0}{\gamma_1} \dot{\mathbf{y}}_{N-1}^2 \right) \end{aligned} \quad (10)$$

SS4 uses

$$\begin{aligned} \mathbf{y}_N &= \mathbf{y}_{N-1} \\ &+ \Delta t \left(\frac{\gamma_0 \gamma_2 \gamma_4 \gamma_6}{\gamma_1 \gamma_3 \gamma_5} \dot{\mathbf{y}}_N + \frac{\gamma_0 \gamma_2 \gamma_4(1 - \gamma_6)}{\gamma_1 \gamma_3 \gamma_5} \dot{\mathbf{y}}_{N-1} + \right. \\ &\quad \left. \frac{\gamma_0 \gamma_2(\gamma_5 - \gamma_4)}{\gamma_1 \gamma_3 \gamma_5} \dot{\mathbf{y}}_{N-1}^1 + \frac{\gamma_0(\gamma_3 - \gamma_2)}{\gamma_1 \gamma_3} \dot{\mathbf{y}}_{N-1}^2 + \frac{\gamma_1 - \gamma_0}{\gamma_1} \dot{\mathbf{y}}_{N-1}^3 \right) \end{aligned} \quad (11)$$

3.1 Prediction

A predictor-corrector approach [24] is used to solve Eqs. (8a)–(8b) in MBDyn. Prediction and correction are two separate, independent, and consecutive phases. The objective of the prediction phase is to determine a tentative value for the solution to start the correction. Here $\dot{\mathbf{y}}_N^{(0)}$ is used to represent the predicted value, and then $\mathbf{y}_N^{(0)}$ can be obtained directly by the integration scheme. The scheme for prediction has minimal effect on the accuracy of the solutions, but it may affect the number of iterations required, as shown in Sect. 5. The closer the predicted value is to the final solution, the fewer iterations are required. Consequently, the criterion for choosing the prediction scheme here is that it should not employ extra

information that was not used in the integration scheme, and that it should have similar accuracy order to the integration scheme.

For the multi-step integrators, $\dot{\mathbf{y}}_N^{(0)}$ is predicted by a second-order scheme, as

$$\dot{\mathbf{y}}_N^{(0)} = \frac{1}{t_N - t_{N-1}} (m_0 \mathbf{y}_{N-1} + m_1 \mathbf{y}_{N-2}) + n_0 \dot{\mathbf{y}}_{N-1} + n_1 \dot{\mathbf{y}}_{N-2} \quad (12)$$

Its local truncation error is defined as

$$\sigma = \dot{\mathbf{y}}(t_N) - n_0 \dot{\mathbf{y}}(t_{N-1}) - n_1 \dot{\mathbf{y}}(t_{N-2}) - \frac{1}{t_N - t_{N-1}} (m_0 \mathbf{y}(t_{N-1}) + m_1 \mathbf{y}(t_{N-2})) \quad (13)$$

Expanding the right-hand side at t_N by Taylor's theorem and letting the local truncation error satisfy $\sigma = O((t_N - t_{N-1})^3)$, i.e., second-order accuracy, yields

$$\begin{aligned} \alpha &= \frac{t_N - t_{N-1}}{t_{N-1} - t_{N-2}} \\ m_0 &= -6\alpha^2(1 + \alpha), & m_1 &= -m_0 \\ n_0 &= (1 + \alpha)(1 + 3\alpha), & n_1 &= \alpha(2 + 3\alpha) \end{aligned} \quad (14)$$

For the single-step integrators, the integration schemes use only the states of the last step. So to make them truly single-step, the constant prediction is used, as

$$\dot{\mathbf{y}}_N^{(0)} = \dot{\mathbf{y}}_{N-1} \quad (15)$$

For the ESDIRKs, if t_N is in the second-stage, using only the states of the last time point, the constant prediction in Eq. (15) is employed. If t_N is in the third to $(s - 1)$ th stage, since the integration scheme in these stages are all second-order accurate, the second-order prediction in Eq. (12) is used. Note that in these cases t_N is the current time and t_{N-1} is the time of the last stage, so $t_N - t_{N-1}$ in Eq. (12) is not the time step size Δt . The accuracy of the last stage has the same order as the overall accuracy, so the second-order ρ_∞ -Bathe method, MSSTC(3, 4, 5), as well as the third-order MSSTH(3), ESDIRK3(2)4L[2]SA and ESDIRK3(2)5L[2]SA still use second-order prediction in the last stage, but for the last stage of the fourth-order MSSTH(4), ESDIRK4(3)6L[2]SA₂ and the fifth-order MSSTH(5), a fourth-order prediction scheme is employed, as

$$\begin{aligned} \dot{\mathbf{y}}_N^{(0)} &= \frac{1}{t_N - t_{N-1}} (m_0 \mathbf{y}_{N-1} + m_1 \mathbf{y}_{N-2} + m_2 \mathbf{y}_{N-3}) \\ &\quad + n_0 \dot{\mathbf{y}}_{N-1} + n_1 \dot{\mathbf{y}}_{N-2} + n_2 \dot{\mathbf{y}}_{N-3} \end{aligned} \quad (16)$$

The parameters are determined by making it fourth-order accurate, as

$$\alpha_1 = \frac{t_N - t_{N-1}}{t_{N-1} - t_{N-2}}, \quad \alpha_2 = \frac{t_N - t_{N-1}}{t_{N-2} - t_{N-3}} \quad (17a)$$

$$m_0 = -\frac{2\alpha_1^2(1 + \alpha_1)}{(\alpha_1 + \alpha_2)^3} (5\alpha_1^3\alpha_2^2 + 8\alpha_1^3\alpha_2 + 10\alpha_1^2\alpha_2^3 + 3\alpha_1^3 + 25\alpha_1^2\alpha_2^2 + 13\alpha_1^2\alpha_2 + 20\alpha_1\alpha_2^3 + 20\alpha_1\alpha_2^2 + 10\alpha_2^3) \quad (17b)$$

$$m_1 = \frac{2(1 + \alpha_1)}{\alpha_1} (5\alpha_1^3\alpha_2^2 + 8\alpha_1^3\alpha_2 + 3\alpha_1^3 - 5\alpha_1^2\alpha_2^3 +) \quad (17c)$$

$$\alpha_1^2 \alpha_2^2 + 4\alpha_1^2 \alpha_2 - 7\alpha_1 \alpha_2^3 - \alpha_1 \alpha_2^2 - 2\alpha_2^3) \\ m_2 = -m_0 - m_1 \quad (17d)$$

$$n_0 = \frac{(1 + \alpha_1)}{(\alpha_1 + \alpha_2)^2} (5\alpha_1^3 \alpha_2^2 + 8\alpha_1^3 \alpha_2 + 3\alpha_1^3 + 11\alpha_1^2 \alpha_2^2) \quad (17e)$$

$$+ 10\alpha_1^2 \alpha_2 + \alpha_1^2 + 7\alpha_1 \alpha_2^2 + 2\alpha_1 \alpha_2 + \alpha_2^2) \\ n_1 = \frac{1}{\alpha_1} (5\alpha_1^3 \alpha_2^2 + 8\alpha_1^3 \alpha_2 + 3\alpha_1^3 + 12\alpha_1^2 \alpha_2^2 + 12\alpha_1^2 \alpha_2 + 2\alpha_1^2 + 9\alpha_1 \alpha_2^2 + 4\alpha_1 \alpha_2 + 2\alpha_2^2) \quad (17f)$$

$$n_2 = \frac{\alpha_2^3 (1 + \alpha_1)}{\alpha_1 (\alpha_1 + \alpha_2)^2} (2\alpha_1 + 2\alpha_2 + 7\alpha_1 \alpha_2 + 5\alpha_1^2 \alpha_2 + 4\alpha_1^2) \quad (17g)$$

In terms of rotations, the orientation and angular velocity of each node used for spatial modelling in MBDyn are stored in the orientation matrix \mathbf{R} , and vector $\boldsymbol{\omega}$, respectively. To predict the orientation at t_N , the Cayley–Gibbs–Rodriguez (CGR) orientation parameters are assumed to be zero at the last point t_{N-1} , namely $\mathbf{g}_{k-1} \equiv \mathbf{0}$, and those of the other previous steps, if needed, are extracted from the respective orientation matrices relative to that at time t_{N-1} , namely $\mathbf{g}_{N-j} = \mathbf{g}(\mathbf{R}_{N-j} \mathbf{R}_{N-1}^T)$, $j \geq 2$. This procedure resembles typical approaches to Lie group integration (e.g., [14]), and was inspired by the spatial interpolation of finite rotations on 1D domains, originally formulated for geometrically exact beam analysis.

The CGR parameter derivatives are evaluated accordingly: $\dot{\mathbf{g}}_{N-1} \equiv \boldsymbol{\omega}_{N-1}$, since $\mathbf{g}_{N-1} \equiv \mathbf{0}$, and $\dot{\mathbf{g}}_{N-j} = \mathbf{G}_{N-j}^{-1} \boldsymbol{\omega}_{N-j}$, $j \geq 2$, where the matrix $\mathbf{G}(\mathbf{g})$ expresses the transformation from the rotation parameter derivatives to the angular velocity, $\boldsymbol{\omega} = \mathbf{G}\dot{\mathbf{g}}$. For the detailed expressions of $\mathbf{g}(\mathbf{R})$ and $\mathbf{G}(\mathbf{g})$ please refer to [24].

The single-step integrators must additionally prepare the auxiliary variables $\dot{\mathbf{g}}_{N-1}^p$ ($1 \leq p \leq r-1$), but they do not have the corresponding CGR parameters \mathbf{g}_{N-1}^p . To simplify the computation, the approximation $\mathbf{g}_{N-1}^p \approx \mathbf{g}_{N-1} \equiv \mathbf{0}$ is used, such that $\dot{\mathbf{g}}_{N-1}^p \equiv \boldsymbol{\omega}_{N-1}^p$. After obtaining the final solutions at t_{N-1} , $\boldsymbol{\omega}_{N-1}^p$ are updated according to the integration scheme. From Eqs. (4a)–(4e), SS2 uses

$$\boldsymbol{\omega}_{N-1}^1 = \frac{1 - \gamma_2}{\gamma_1} \boldsymbol{\omega}_{N-2} + \frac{\gamma_2}{\gamma_1} \boldsymbol{\omega}_{N-1} - \frac{1 - \gamma_1}{\gamma_1} \boldsymbol{\omega}_{N-2}^1 \quad (18)$$

SS3 uses

$$\boldsymbol{\omega}_{N-1}^1 = \frac{1 - \gamma_4}{\gamma_3} \boldsymbol{\omega}_{N-2} + \frac{\gamma_4}{\gamma_3} \boldsymbol{\omega}_{N-1} - \frac{1 - \gamma_3}{\gamma_3} \boldsymbol{\omega}_{N-2}^1 \quad (19a)$$

$$\boldsymbol{\omega}_{N-1}^2 = \frac{1 - \gamma_2}{\gamma_1} \boldsymbol{\omega}_{N-2}^1 + \frac{\gamma_2}{\gamma_1} \boldsymbol{\omega}_{N-1}^1 - \frac{1 - \gamma_1}{\gamma_1} \boldsymbol{\omega}_{N-2}^2 \quad (19b)$$

SS4 uses

$$\boldsymbol{\omega}_{N-1}^1 = \frac{1 - \gamma_6}{\gamma_5} \boldsymbol{\omega}_{N-2} + \frac{\gamma_6}{\gamma_5} \boldsymbol{\omega}_{N-1} - \frac{1 - \gamma_5}{\gamma_5} \boldsymbol{\omega}_{N-2}^1 \quad (20a)$$

$$\boldsymbol{\omega}_{N-1}^2 = \frac{1 - \gamma_4}{\gamma_3} \boldsymbol{\omega}_{N-2}^1 + \frac{\gamma_4}{\gamma_3} \boldsymbol{\omega}_{N-1}^1 - \frac{1 - \gamma_3}{\gamma_3} \boldsymbol{\omega}_{N-2}^2 \quad (20b)$$

$$\boldsymbol{\omega}_{N-1}^3 = \frac{1 - \gamma_2}{\gamma_1} \boldsymbol{\omega}_{N-2}^2 + \frac{\gamma_2}{\gamma_1} \boldsymbol{\omega}_{N-1}^2 - \frac{1 - \gamma_1}{\gamma_1} \boldsymbol{\omega}_{N-2}^3 \quad (20c)$$

Certainly, other derivatives involved in $\dot{\mathbf{y}}_{N-1}$ must be updated in the same way before computing at t_N . The auxiliary variables of SS3 and SS4 may be complex numbers, so their real and imaginary parts are computed and stored separately. Due to the approximation of rotations, these single-step integrators are no longer exactly equivalent to the corresponding multi-step methods, but their solutions in numerical experiments are still very close, as shown in Sect. 5. This indicates that the approximation does not cause any obvious reduction in accuracy.

Having obtained the required \mathbf{g}_{N-j} , $\dot{\mathbf{g}}_{N-j}$ ($j \geq 1$) and \mathbf{g}_{N-1}^p ($1 \leq p \leq r-1$), the CGR parameters and their derivatives at time t_N , $\mathbf{g}_N^{(0)}$ and $\dot{\mathbf{g}}_N^{(0)}$, are predicted using the previously illustrated schemes. Then, the predicted orientation matrix and angular velocity are computed as $\mathbf{R}_\Delta = \mathbf{R}(\mathbf{g}_N^{(0)})$, $\mathbf{R}_N^{(0)} = \mathbf{R}_\Delta \mathbf{R}_{N-1}$ and $\mathbf{G}_\Delta = \mathbf{G}(\mathbf{g}_N^{(0)})$, $\boldsymbol{\omega}^{(0)} = \mathbf{G}_\Delta \dot{\mathbf{g}}_N^{(0)}$.

Of course, the relative rotation between the involved time steps is assumed limited to avoid the indeterminacies and singularities inherent in all three-parameter parameterizations, specifically those related to the CGR parameters (the magnitude of the relative rotations must be significantly smaller than π). Such an assumption is considered acceptable when the time step of the integration is dictated by accuracy requirements.

3.2 Correction

After the prediction phase, by a Newton-like iteration method, \mathbf{y}_N and $\dot{\mathbf{y}}_N$ are corrected according to

$$\left[\frac{\partial \mathbf{y}_N}{\partial \dot{\mathbf{y}}_N} \mathbf{r}_y \left(\mathbf{y}_N^{(l)}, \dot{\mathbf{y}}_N^{(l)}, t_N \right) + \mathbf{r}_{\dot{y}} \left(\mathbf{y}_N^{(l)}, \dot{\mathbf{y}}_N^{(l)}, t_N \right) \right] \Delta \dot{\mathbf{y}} \quad (21a)$$

$$= -\mathbf{r} \left(\mathbf{y}_N^{(l)}, \dot{\mathbf{y}}_N^{(l)}, t_N \right) \quad (21b)$$

$$\dot{\mathbf{y}}_N^{(l+1)} = \dot{\mathbf{y}}_N^{(l)} + \Delta \dot{\mathbf{y}} \quad (21c)$$

where \mathbf{r}_y and $\mathbf{r}_{\dot{y}}$ are the partial derivatives of \mathbf{r} with respect to \mathbf{y} and $\dot{\mathbf{y}}$, respectively; $\partial \mathbf{y}_N / \partial \dot{\mathbf{y}}_N = \beta_0 \Delta t$ for the multi-step integrators, $\partial \mathbf{y}_N / \partial \dot{\mathbf{y}}_N = \gamma_0 \gamma_2 \Delta t / \gamma_1$ for SS2, $\partial \mathbf{y}_N / \partial \dot{\mathbf{y}}_N = \gamma_0 \gamma_2 \gamma_4 \Delta t / (\gamma_1 \gamma_3)$ for SS3, $\partial \mathbf{y}_N / \partial \dot{\mathbf{y}}_N = \gamma_0 \gamma_2 \gamma_4 \gamma_6 \Delta t / (\gamma_1 \gamma_3 \gamma_5)$ for SS4, and $\partial \mathbf{y}_N / \partial \dot{\mathbf{y}}_N = \gamma \Delta t$ for the ESDIRKs; l denotes the number of iterations, and $l = 0$ at the initial.

In the correction phase, at each iteration, $\mathbf{g}_N^{(l)}$ and $\dot{\mathbf{g}}_N^{(l)}$ are obtained through Eqs. (21a)–(21c). The orientation is recast as $\mathbf{R}_N^{(l)} = \mathbf{R}_\Delta \mathbf{R}_N^{(0)}$ with $\mathbf{R}_\Delta = \mathbf{R}(\mathbf{g}_N^{(l)})$. The angular velocity is expressed as $\boldsymbol{\omega}_N^{(l)} = \mathbf{G}_\Delta \dot{\mathbf{g}}_N^{(l)} + \mathbf{R}_\Delta \boldsymbol{\omega}_N^{(0)}$ with $\mathbf{G}_\Delta = \mathbf{G}(\mathbf{g}_N^{(l)})$.

The final solution at t_N is obtained when the prescribed convergence condition is satisfied. Then the procedure moves on to the solution for the next time point.

3.3 Generalization and extension

All previously described methods, which are summarized in Fig. 1, are available in MB-Dyn's public source code repository.² Other time integration schemes can be easily added using the provided class templates.

²<https://public.gitlab.polimi.it/DAER/mbdyn>, in the “integrators” development branch, to be merged soon with the main development branch.

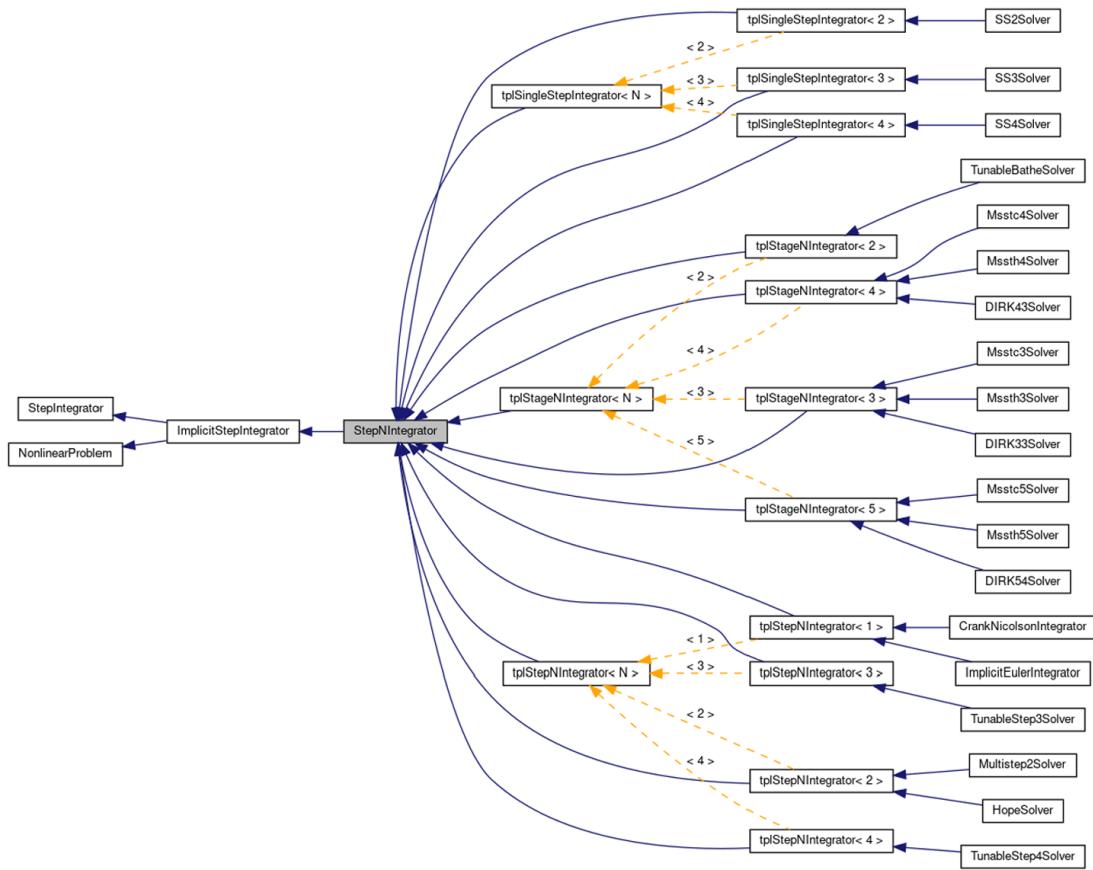


Fig. 1 Inheritance tree of the `StepNIntegrator` class, generated using Doxygen

To define a new integrator that belongs to the families of linear multi-step, equivalent single-step, or multi-stage methods, one simply needs to derive a new class from the corresponding template class, define the number of steps or stages (an integer template value), and overload the virtual methods that provide the coefficients for the prediction of state and derivative.

The implementation of these methods in MBDyn granted the possibility to evaluate their performance when applied to non-trivial, general-purpose multibody system dynamics problems. Some examples taken from the literature are investigated in Sect. 5.

4 Linear analysis

The dissipation and dispersion accuracy as well as the degree of algorithmic dissipation of the employed methods are compared in this section considering the single degree-of-freedom problem $\ddot{x} + \omega^2 x = 0$, an undamped oscillator. The numerical solution of the problem at time t_k can be expressed as

$$x_k = e^{-\bar{\xi}\bar{\omega}t_k} (c_1 \cos(\bar{\omega}_d t_k) + c_2 \sin(\bar{\omega}_d t_k)), \quad \bar{\omega}_d = \bar{\omega} \sqrt{1 - \bar{\xi}^2} \quad (22)$$

where c_1 and c_2 are constants determined by the initial conditions, and $\bar{\omega}$ and $\bar{\xi}$ are the numerical natural frequency and damping ratio, respectively. Here $\bar{\xi}$ and $(\bar{T} - T)/T = \omega/\bar{\omega} - 1$ are known as the amplitude decay ratio and period elongation ratio, respectively. They are

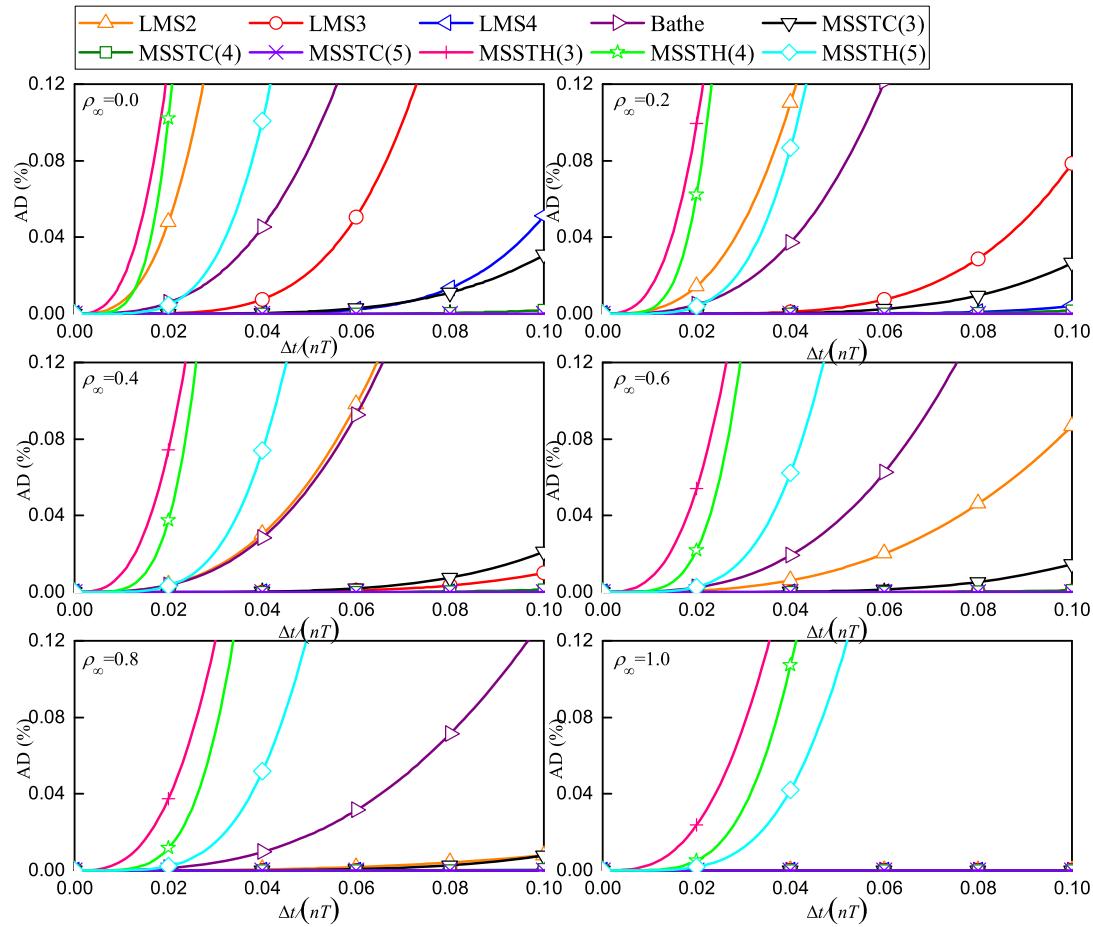


Fig. 2 Percentage amplitude decay of the methods with tunable algorithmic dissipation

typically used to measure the dissipation and dispersion accuracy in the low-frequency domain. They can be obtained from the characteristic roots of the method, as in [37]. Besides, the spectral radius of the method is used to measure the degree of algorithmic dissipation.

Figures 2, 3 and 4 summarize the percentage amplitude decay (AD(%)), percentage period elongation (PE(%)) and spectral radius (ρ) of methods with tunable algorithmic dissipation, respectively. Considering the intrinsic spectral equivalence of the single-step methods and the corresponding multi-step methods, only the results of the multi-step methods are shown. Figures 5, 6 and 7 show AD(%), PE(%) and ρ of the higher-order ESDIRKs with $\rho_\infty = 0.0$. Because the s -stage ESDIRKs perform $s - 1$ implicit stages per step, to compare the results under similar computational cost, the abscissa is set as $\Delta t/(nT)$ in the figures, where $n = 1$ for the multi-step methods, and $n = s - 1$ for the s -stage ESDIRKs.

The employed methods all have A-stability and can provide algorithmic dissipation. Among them, LMS2, LMS3, LMS4, Bathe, MSSTC(3, 4, 5) are second-order accurate, and other ESDIRKs have higher-order accuracy. From the comparison, one can observe that the second-order methods, especially LMS3, LMS4, MSSTC(3, 4, 5), are really superior in preserving the frequency content. They show very small numerical damping ratios and spectral radii very close to 1 when $\Delta t/(nT) \leq 0.1$.

As the number of steps or stages used increases, both dissipation and dispersion accuracy improve for second-order methods, so LMS4 and MSSTC(5) have better accuracy, and can retain more frequency content. As ρ_∞ grows, these second-order methods also show higher

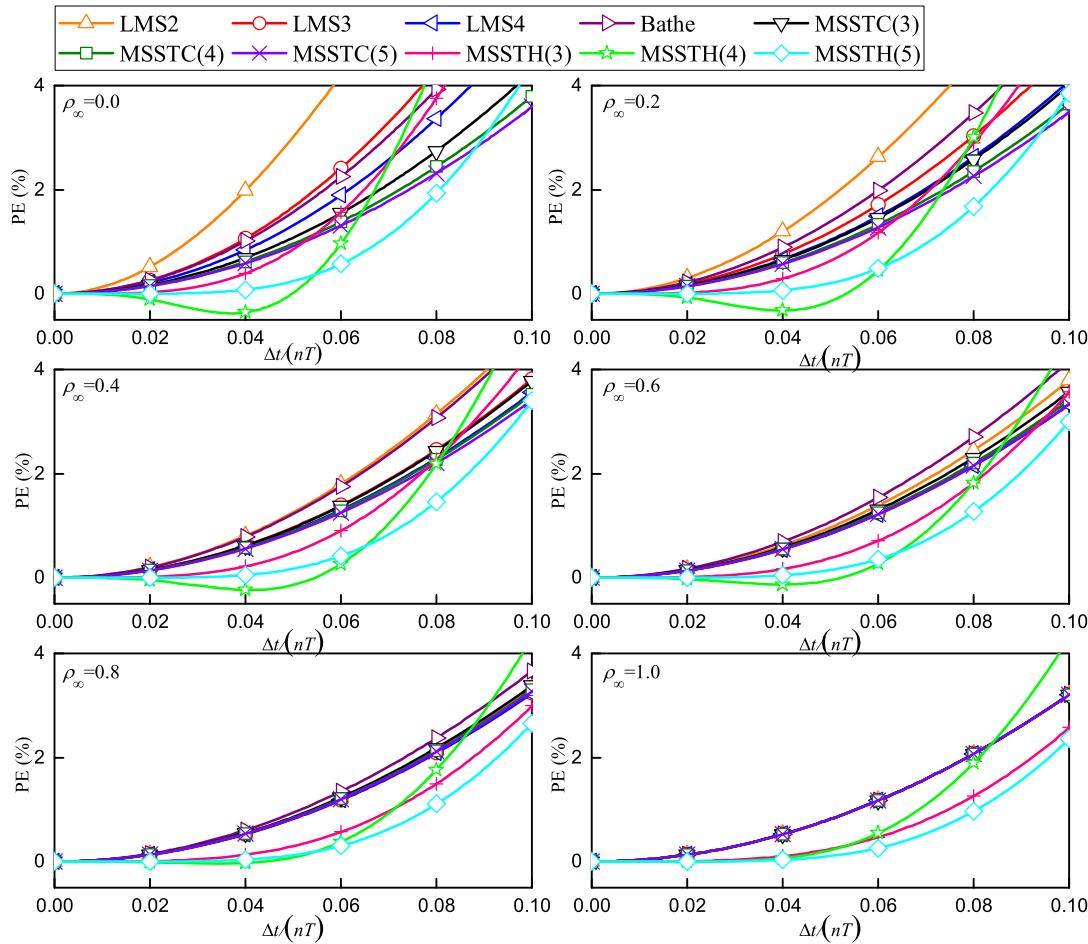


Fig. 3 Percentage period elongation of the methods with tunable algorithmic dissipation

accuracy, and when $\rho_\infty = 1.0$, they all have the same characteristics as the trapezoidal rule. Compared to the multi-step methods, the second-order multi-stage ones show better filtering ability for high-frequency content, since their spectral radii drop to ρ_∞ faster and earlier.

Compared to the second-order methods, the higher-order ESDIRKs exhibit higher dispersion accuracy, which gives them some accuracy advantage in transient simulations. On the other hand, the excessive algorithmic dissipation of MSSTH(3, 4, 5) and ESDIRK3(2)4L-[2]SA dissipates most of the frequency content in long-term simulations. With $\rho_\infty = 0.0$, MSSTH(5), ESDIRK3(2)5L[2]SA and ESDIRK4(3)6L[2]SA₂ show very small period elongation for $\Delta t / (nT) \leq 0.05$, whereas MSSTH(4) exhibits period shortening. MSSTH(3) with $\rho_\infty = 0.0$ has spectral characteristics almost identical to those of ESDIRK3(2)4L[2]SA. Even though MSSTH(3, 4, 5) have tunable algorithmic dissipation, their accuracy cannot be improved when ρ_∞ increases, and when $\rho_\infty = 1.0$ they exhibit unexpected algorithmic dissipation at intermediate frequencies. For these reasons, a small ρ_∞ , like 0.0, is recommended for MSSTH(3, 4, 5) to improve stability of the algorithm.

From the linear analysis, it appears that the second-order methods are very effective in preserving the amplitude, while the higher-order methods have better phase accuracy. Among second-order methods, the use of more steps or stages or a larger ρ_∞ helps to improve the dissipation and dispersion accuracy. Multi-step methods can retain more frequency content than multi-stage ones with the same ρ_∞ . Except for ESDIRK4(3)6L[2]SA₂

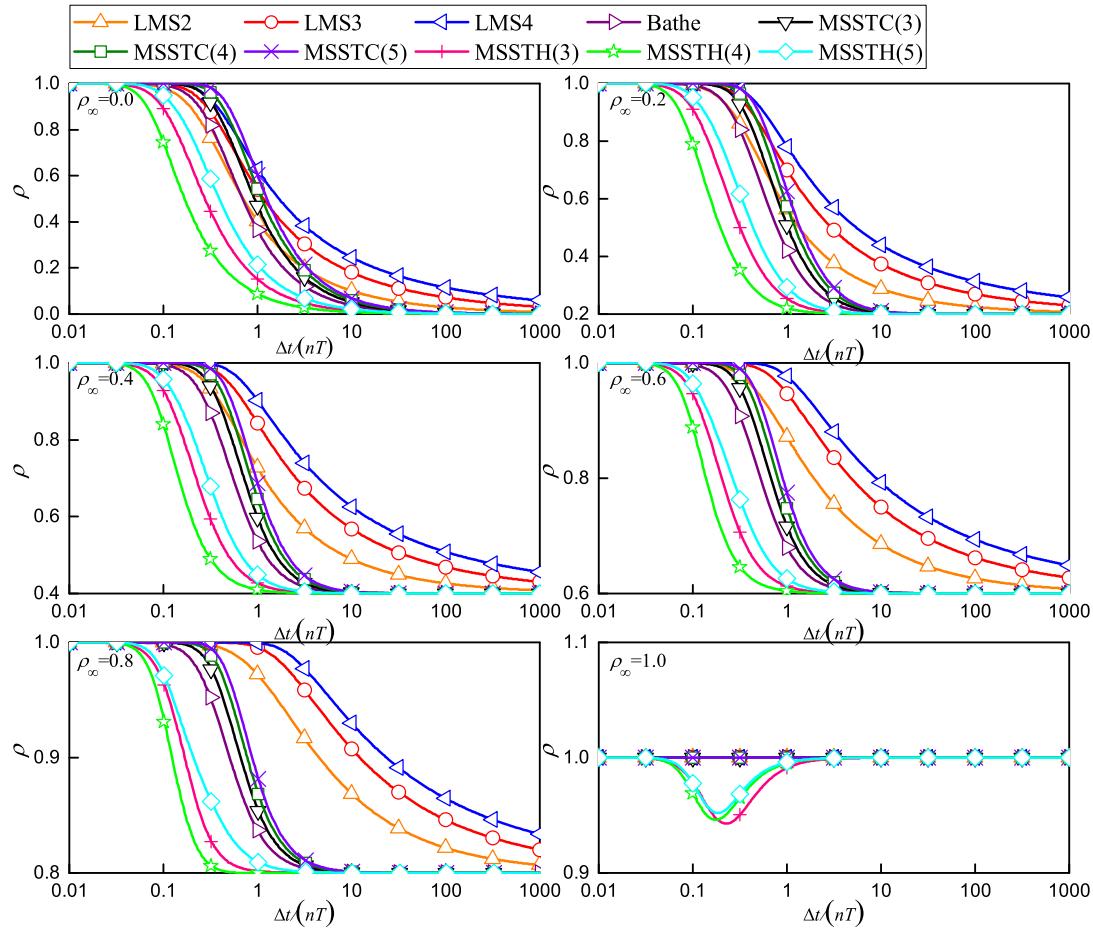
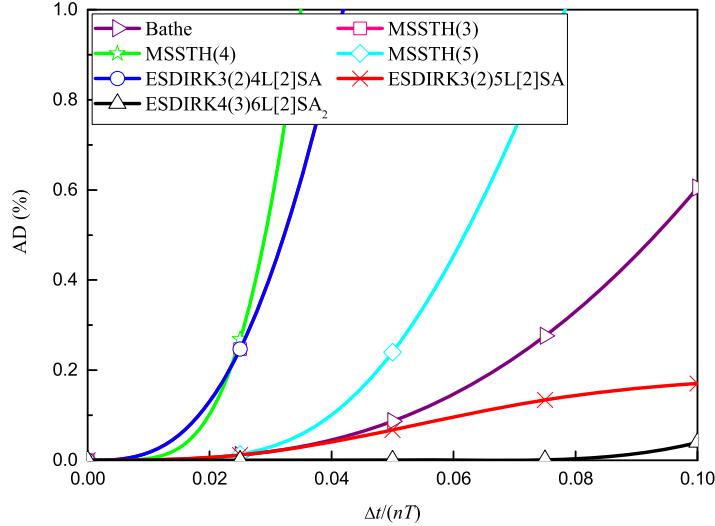


Fig. 4 Spectral radius of the methods with tunable algorithmic dissipation

Fig. 5 Percentage amplitude decay of the higher-order methods with $\rho_\infty = 0.0$



and ESDIRK3(2)5L[2]SA, the employed other higher-order methods show excessive algorithmic dissipation in the low-frequency domain.

Fig. 6 Percentage period elongation of the higher-order methods with $\rho_\infty = 0.0$

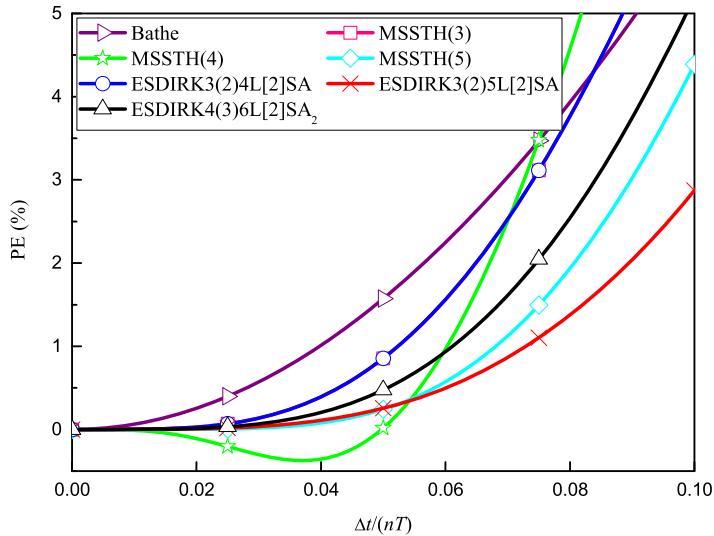
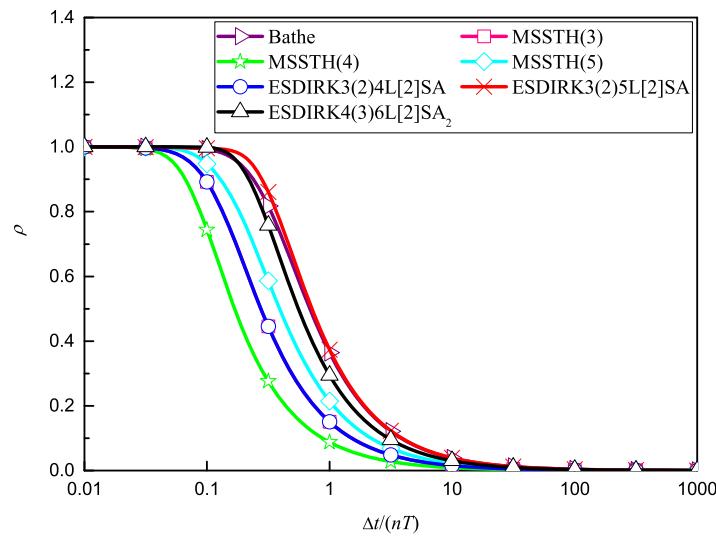


Fig. 7 Spectral radius of the higher-order methods with $\rho_\infty = 0.0$



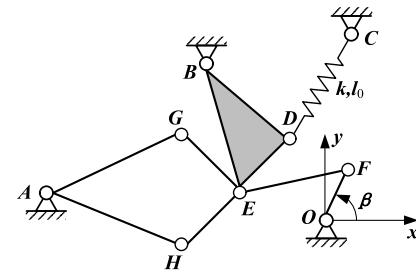
5 Numerical experiments

The performance of the previously discussed integrators is illustrated in this Section by solving some benchmark problems in MBDyn.³ Two values of ρ_∞ , 0.0 and 0.6, are used for second-order methods, and $\rho_\infty = 0.0$ is used in MSSTH(3, 4, 5). In all examples, the integrators use the same $\Delta t/n$ to predict the results under comparable computational cost. The results of the second-order integrators with $\rho_\infty = 0.0$, the second-order integrators with $\rho_\infty = 0.6$, and the higher-order integrators, are presented separately. Table 1 lists the line and symbol of different integrators used in the result figures in this Section. The accuracy and stability of the numerical results as well as the calculation efficiency of the methods, are discussed.

³The input files for MBDyn are publicly available in the examples folder of the software distribution, <https://public.gitlab.polimi.it/DAER/mbdyn>.

Table 1 Line and symbol of different methods used in the figures in Sect. 5

Method	Line	Symbol	Method	Line	Symbol
Reference	—	×	LMS2	—	△
LMS3	—	○	LMS4	—	◀
SS2	—	+	SS3	—	★
SS4	—	◇	Bathe	—	▷
MSSTC(3)	—	▽	MSSTC(4)	—	□
MSSTC(5)	—	×			
Reference	—	×	MSSTH(3)	—	△
MSSTH(4)	—	○	MSSTH(5)	—	◀
ESDIRK3(2)4L[2]S	—	+	ESDIRK3(3)5L[2]SA	—	★
ESDIRK4(3)6L[2]SA ₂	—	◇			

Fig. 8 Andrew's squeezer mechanism (adapted from [29])

5.1 Andrew's squeezer mechanism

Problem description Andrew's squeezer mechanism [29], as shown in Fig. 8, is a planar system composed of seven rigid links. The coordinates of noteworthy points in the local reference frame, the mass, and the moment of inertia of each part are listed in Table 2. The origin of each local reference frame is placed in the first point with the name of each link. For links with only two points, the local x -axis is along the line connecting the points. For the link $E-B-D$, the local y -axis is aligned with $E-B$, pointing towards B . The coordinates of the center of mass (x_C, y_C) are described in the local reference frame, and the rotational inertia I_z of each body is expressed about its centre of mass. In the global reference frame Oxy , the coordinates of points A, B, C are $(-0.06934 \text{ m}, -0.00227 \text{ m})$, $(-0.03635 \text{ m}, 0.03273 \text{ m})$ and $(0.01400 \text{ m}, 0.07200 \text{ m})$, respectively. The point D is connected to the point C by a spring, whose stiffness characteristic is $k = 4530 \text{ N/m}$ and whose natural length is $l_0 = 0.07785 \text{ m}$. The link $O-F$ is driven with a constant torque $T = 0.033 \text{ N} \cdot \text{m}$, starting from an initial angle $\beta_0 = -0.0620 \text{ rad}$. Gravity is not considered.

Numerical results This example is used to check how the methods employed can preserve the mechanical energy of the system. The total energy balance equation of the system can be expressed as

$$\Delta E = E - E_0 - T(\beta - \beta_0) \quad (23)$$

since the torque T is constant, where E collects the kinetic and potential energy of the system. Without physical damping, ΔE should be zero throughout the simulation. With $\Delta t/n = 10^{-4} \text{ s}$, Figs. 9 and 10 show, respectively, the angular velocity ω of bar $O-F$, and

Table 2 Coordinates of the points in the local reference system, mass and inertia properties of the links

Link	Point	x (m)	y (m)	x_C (m)	y_C (m)	Mass (kg)	I_z ($\text{kg} \cdot \text{m}^2$)
$O-F$	O	0.0	0.0	0.00092	0.0	0.04325	2.194×10^{-6}
	F	0.007	0.0				
$E-F$	E	0.0	0.0	0.01650	0.0	0.00365	4.410×10^{-7}
	F	0.028	0.0				
$H-E$	H	0.0	0.0	0.00579	0.0	0.00706	5.667×10^{-7}
	E	0.02	0.0				
$G-E$	G	0.0	0.0	0.00579	0.0	0.00706	5.667×10^{-7}
	E	0.02	0.0				
$A-G$	A	0.0	0.0	0.02308	0.00916	0.07050	1.169×10^{-5}
	G	0.04	0.0				
$A-H$	A	0.0	0.0	0.01228	-0.00449	0.05498	1.912×10^{-5}
	H	0.04	0.0				
$E-B-D$	E	0.0	0.0	0.01043	0.01626	0.02373	5.255×10^{-6}
	B	0.0	0.035				
	D	0.02	0.017				

ΔE as obtained from the numerical solutions within the $[0, 0.05]$ s interval. Figures 11 and 12 show the same results but with a smaller step size $\Delta t/n = 10^{-5}$ s. The results given by the second-order methods with $\rho_\infty = 0.0$ and 0.6, as well as the higher-order methods with $\rho_\infty = 0.0$, are presented separately. The reference solutions in Figs. 9 and 11 are obtained by LMS4 with $\rho_\infty = 0.6$ and $\Delta t = 10^{-6}$ s.

From Fig. 9, with $\Delta t/n = 10^{-4}$ s, the results computed using MSSTH(5) with $\rho_\infty = 0.0$ depart from the reference solution after about 0.01 s, and those obtained using MSSTH(3, 4) with $\rho_\infty = 0.0$, ESDIRK3(2)4L[2]SA, and Bathe with $\rho_\infty = 0.6$ show observable differences. From Fig. 10, in this case, the results of ΔE given by MSSTC(5) with $\rho_\infty = 0.0$, MSSTH(3, 4) with $\rho_\infty = 0.0$, and ESDIRK3(2)4L[2]SA, show an increasing trend, indicating that their results are likely to become unstable over time. LMS2 and SS2 with $\rho_\infty = 0.6$, LMS3, SS3, LMS4, SS4, MSSTC(3) and MSSTC(5) with $\rho_\infty = 0.6$, can mostly preserve energy, although with oscillations, while LMS2 and SS2 with $\rho_\infty = 0.0$, Bathe, MSSTC(4) and the remaining higher-order integrators show an obvious drop in energy.

With $\Delta t/n = 10^{-5}$ s, Fig. 11 shows that all methods can predict accurate results of ω . However, from Fig. 12, it is clear that the energy instability, i.e. $\Delta E > 0$ [5], is still observed for some higher-order integrators, including MSSTH(3, 4) with $\rho_\infty = 0.0$, ESDIRK3(2)4L[2]SA, ESDIRK4(3)6L[2]SA₂. MSSTH(5) with $\rho_\infty = 0.0$ exhibits obvious energy drop. The second-order integrators perform well with the smaller step size.

From this example we can conclude that most of the higher-order integrators employed have worse stability in this type of problem, and are really not recommended for energy-conserving purpose. Among the second-order integrators, the linear analysis in Sect. 4 illustrates that they can preserve more modes as the number of steps or stages used increases, but the multi-stage methods do not follow this rule here. This is because they use the trapezoidal rule from the second to the n th stage, and the non-dissipative trapezoidal rule is likely to give unreliable or unstable results for nonlinear problems [33]. Therefore, the multi-step

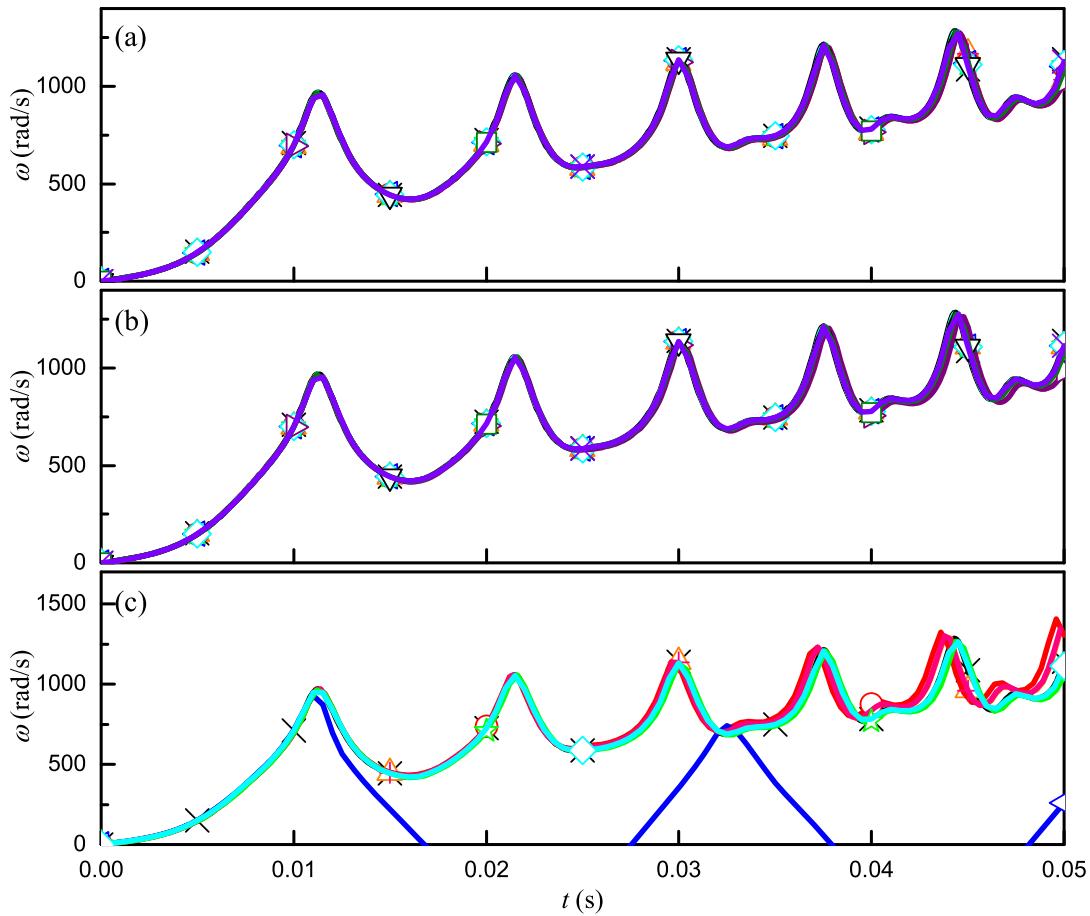


Fig. 9 Angular velocity ω of bar $O-F$ within $[0, 0.05]$ s using $\Delta t/n = 10^{-4}$ s **(a)** Second-order methods with $\rho_\infty = 0.0$; **(b)** Second-order methods with $\rho_\infty = 0.6$; **(c)** Higher-order methods with $\rho_\infty = 0.0$

and single-step integrators, especially LMS3, SS3, LMS4 and SS4 with a large ρ_∞ , such as 0.6, are better candidates in terms of energy conservation for general problems.

5.2 Flexible four-bar mechanism

Problem description Some of the benchmark problems for flexible mechanisms proposed in [4] are solved. Figure 13 shows the configuration of a flexible four-bar mechanism. Bars $A-B$, $B-C$, $C-D$ and the ground are connected through revolute joints. The initial angles between them are all 90 deg. The bars' lengths are $L_1 = L_3 = 0.12$ m and $L_2 = 0.24$ m. The rotation axis of the revolute joint at point C is rotated by +5 deg about the y direction, to simulate an assembly defect that would lock the mechanism if the bars were rigid. The inertia and stiffness properties of the bars are listed in Table 3. Each bar is modeled in MBDyn using 5 three-node beam elements [15]. The angular velocity of the bar $A-B$ at point A with respect to the frame is prescribed as $\Omega = 0.6$ rad/s during the simulation.

Numerical results The simulation was run in the interval $[-2T, 12]$ s, where $T = 2\pi/\Omega$ performs about three complete cycles of the bar $A-B$. With $\Delta t/n = 4 \times 10^{-3}$ s, Figs. 14 and 15 respectively show the rotation angle θ and the angular velocity ω_1 about the x axis at the tip of bar $B-C$ at point C within $[0, 12]$ s. Figure 16 plots the results of ω_1 with a smaller step size $\Delta t = 1 \times 10^{-3}$ s.

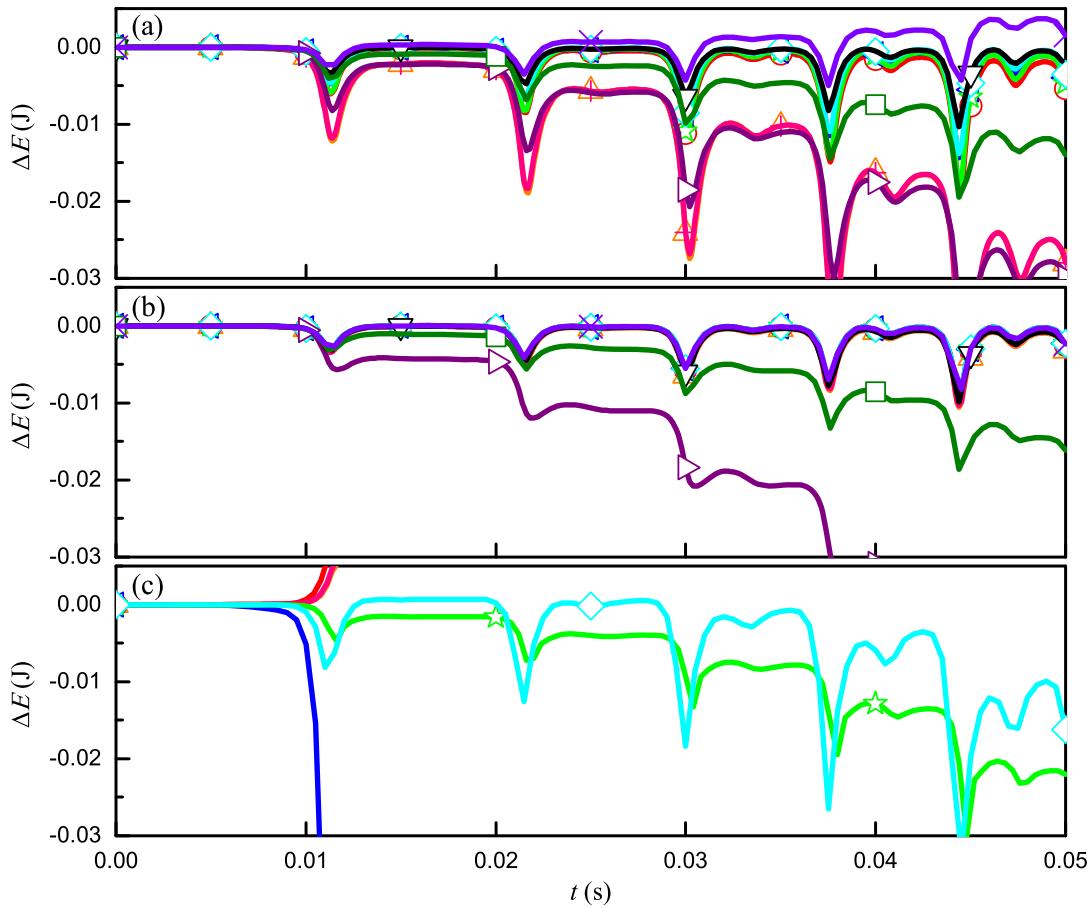


Fig. 10 Energy balance ΔE within $[0, 0.05]$ s using $\Delta t/n = 10^{-4}$ s **(a)** Second-order methods with $\rho_\infty = 0.0$; **(b)** Second-order methods with $\rho_\infty = 0.6$; **(c)** Higher-order methods with $\rho_\infty = 0.0$

Table 3 Inertia and stiffness properties of bars *A-B*, *B-C* and *C-D*

	Bar <i>A-B</i>	Bar <i>B-C</i>	Bar <i>C-D</i>
Mass per unit span m (kg/m)	1.9968	0.4992	1.9968
Moments of inertia per unit span J_1 (mg · m)	85.1968	5.3248	85.1968
Moments of inertia per unit span J_2 (mg · m)	42.5984	2.6624	42.5984
Moments of inertia per unit span J_3 (mg · m)	42.5984	2.6624	42.5984
Axial stiffness EA (MN)	52.9920	13.2480	52.9920
Shearing stiffness GA_Y (MN)	16.8803	4.2201	16.8803
Shearing stiffness GA_Z (MN)	16.8803	4.2201	16.8803
Torsional stiffness GJ (N · m ²)	733.488	45.843	733.488
Bending stiffness EJ_Y (N · m ²)	1130.50	70.656	1130.50
Bending stiffness EJ_Z (N · m ²)	1130.50	70.656	1130.50

From Fig. 14 it is seen that the rotation angles θ computed by the methods used agree very well with each other. However, as shown in Figs. 15 and 16, since the angular velocity changes rapidly at certain moments, high-frequency oscillations can be observed, and the

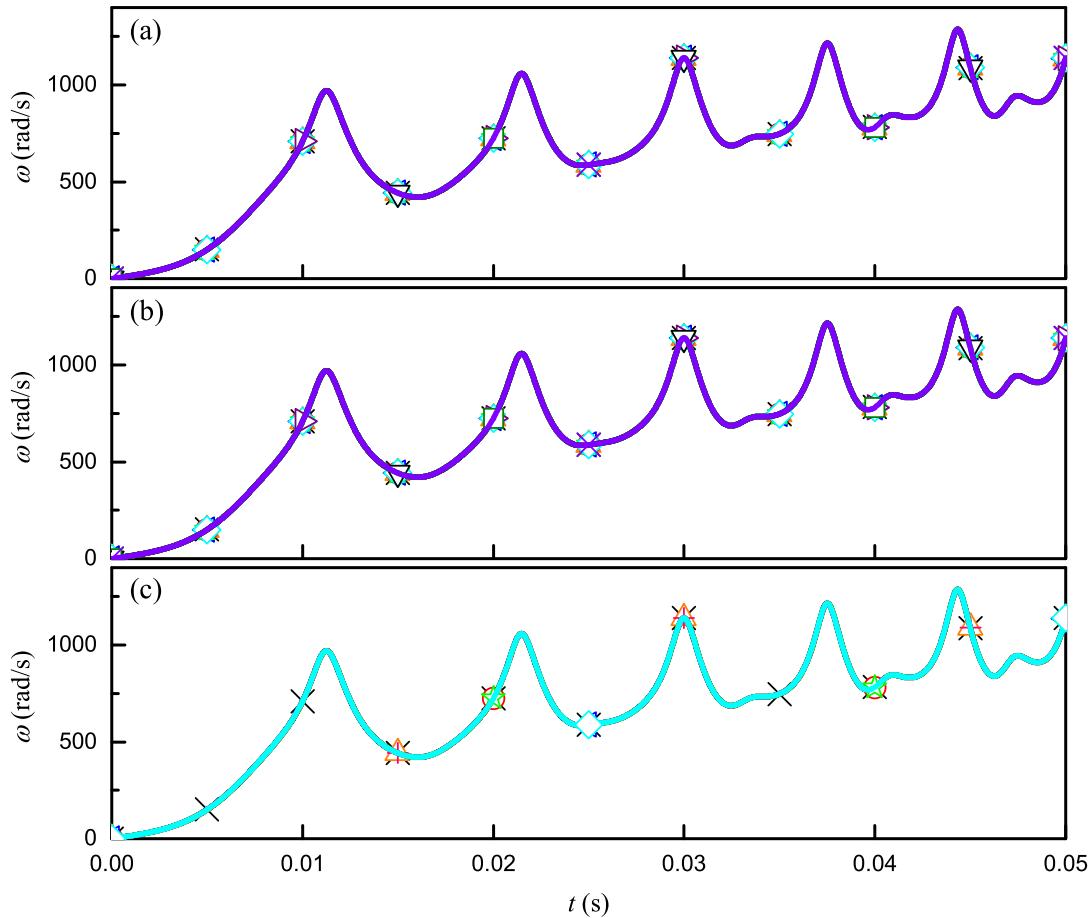


Fig. 11 Angular velocity ω of bar $O\text{-}F$ within $[0, 0.05]$ s using $\Delta t/n = 10^{-5}$ s **(a)** Second-order methods with $\rho_\infty = 0.0$; **(b)** Second-order methods with $\rho_\infty = 0.6$; **(c)** Higher-order methods with $\rho_\infty = 0.0$

observable differences in the integrators employed are how fast they filter the oscillations. As shown in Figs. 15 and 16, the oscillations become more pronounced with a smaller time step size or a larger ρ_∞ . LMS4 and SS4 with $\rho_\infty = 0.6$ and $\Delta t = 1 \times 10^{-3}$ s exhibit the most significant oscillations. This example shows the importance of algorithmic dissipation for problems containing high-frequency pollution, which often appears in the solutions of velocities, accelerations and forces. For such problems, Bathe and the higher-order integrators, which have strong algorithmic dissipation from linear analysis in Sect. 4, with $\rho_\infty = 0.0$ and a suitable Δt , not too small, are recommended.

5.3 Lateral buckling of a thin beam

Problem description Figure 17 shows the configuration of a beam actuated by a crank-link mechanism. The beam is clamped at one end; the other end is connected to the link by a spherical joint. To simulate an initial imperfection, the plane of the crank-link mechanism is offset from the plane of the beam by $d = 0.1$ mm in the y direction. The end of the beam and the spherical joint are rigidly connected. The link, crank, and ground are connected via revolute joints. The lengths are $L = 1$ m, $L_l = 0.25$ m and $L_c = 0.05$ m. The inertia and stiffness properties of the crank, link and beam are listed in Table 4. The beam is meshed with 5 three-node beam elements, and both the link and the crank are modelled with 1 three-

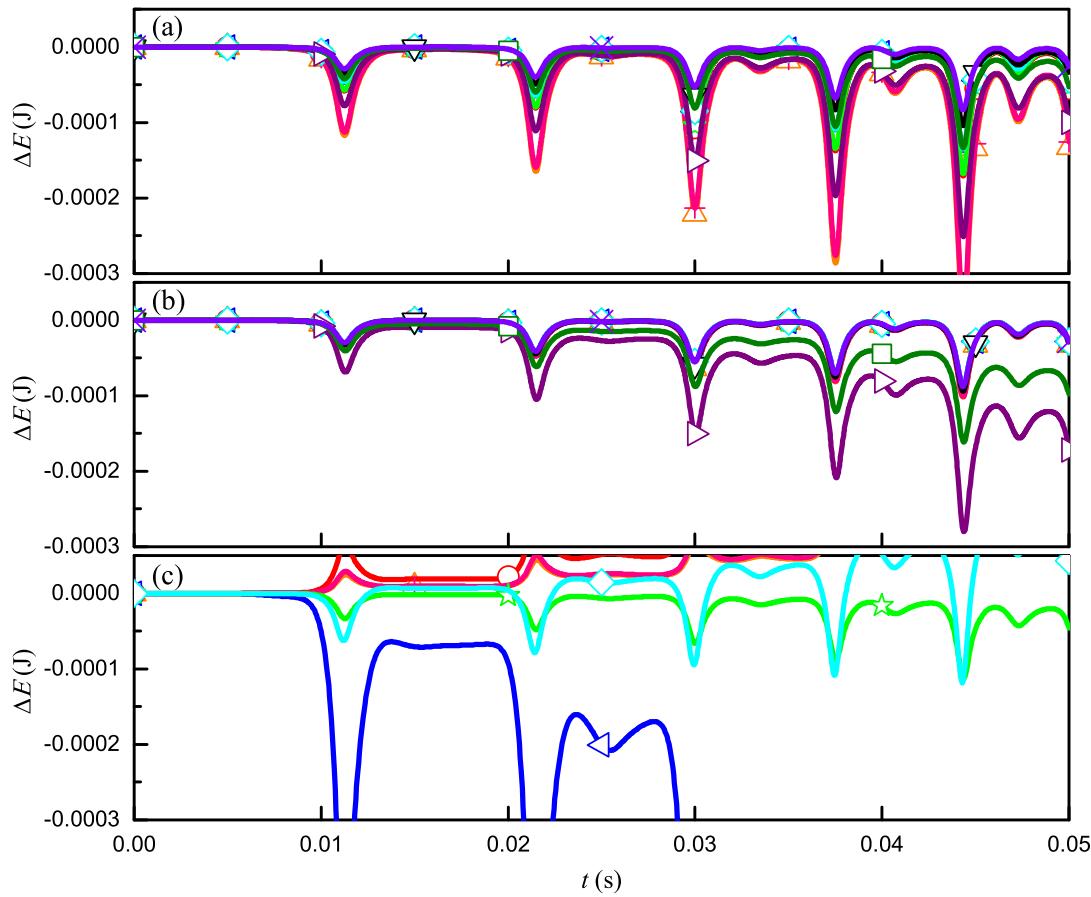
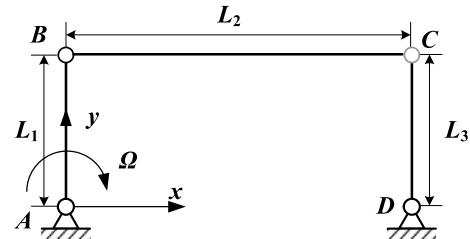


Fig. 12 Energy balance ΔE within $[0, 0.05]$ s using $\Delta t/n = 10^{-5}$ s **(a)** Second-order methods with $\rho_\infty = 0.0$; **(b)** Second-order methods with $\rho_\infty = 0.6$; **(c)** Higher-order methods with $\rho_\infty = 0.0$

Fig. 13 Flexible four-bar mechanism (adapted from [4])



node beam element each. The rotation angle of the crank is prescribed as

$$\phi = \begin{cases} \frac{\pi}{2} \left(1 - \cos \frac{\pi t}{T} \right), & t \leq T \\ \pi, & t > T \end{cases} \quad (24)$$

where $T = 0.4$ s.

Numerical results The simulation was run in the interval $[0, 0.8]$ s using $\Delta t/n = 10^{-3}$ s. The rotation angle θ and angular velocity ω_1 about the x axis, and the shear force F_3 along the z axis at the mid-span of the beam are summarized in Figs. 18, 19 and 20, respectively. Driven by the crank-link mechanism, the beam buckled laterally rather quickly, before the

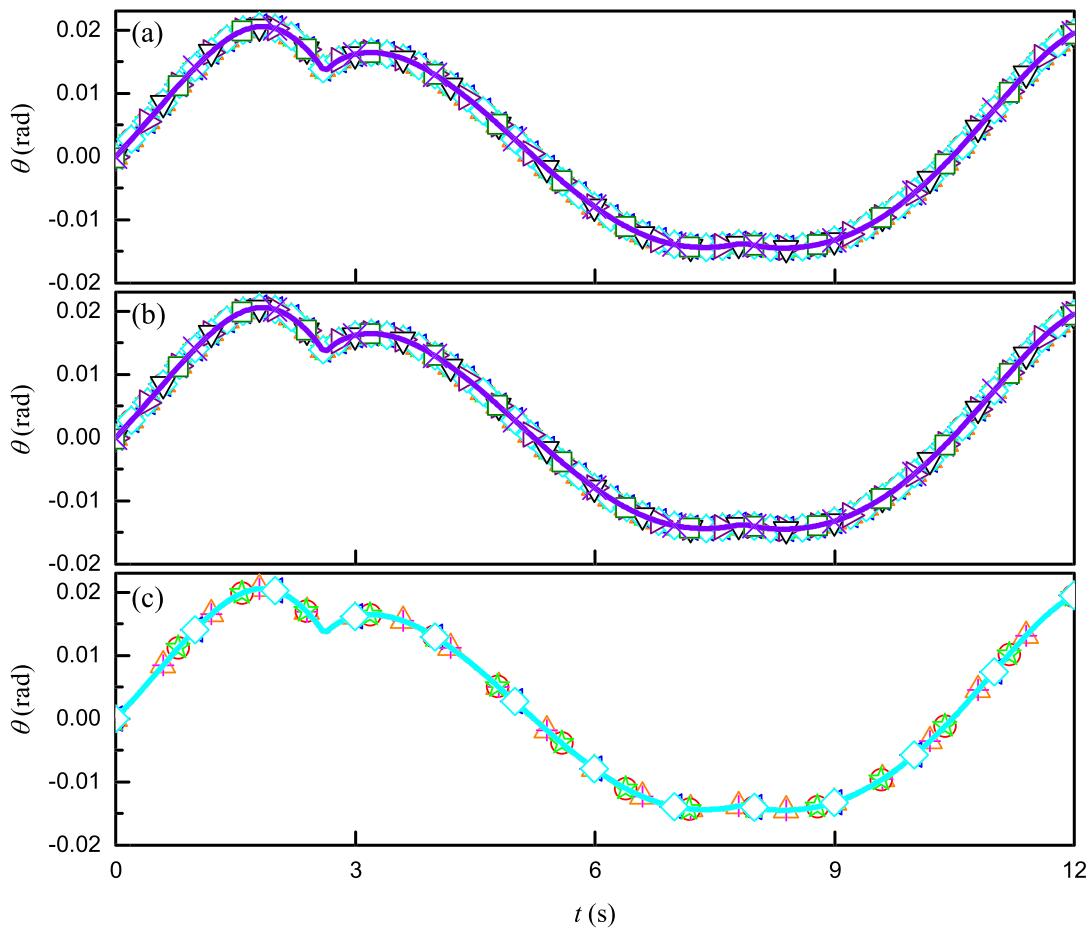


Fig. 14 Rotation angle θ about the x axis at the tip of bar $B-C$ at point C within $[0, 12]$ s using $\Delta t/n = 4 \times 10^{-3}$ s (a) Second-order methods with $\rho_\infty = 0.0$; (b) Second-order methods with $\rho_\infty = 0.6$; (c) Higher-order methods with $\rho_\infty = 0.0$

crank reached the uppermost position, and then began to oscillate rapidly. In this case, the higher-order MSSTH(5) with $\rho_\infty = 0.0$, and ESDIRK3(3)5L[2]SA failed to complete the simulation.

As shown in Figs. 18–20, the solutions of the employed methods almost overlap in the $[0, 0.4]$ s interval. However, in the subsequent free oscillations, the solutions obtained with a few methods, including LMS2 and SS2 with $\rho_\infty = 0.0$, MSSTH(3, 4), ESDIRK3(2)4L[2]SA, exhibit a significant amplitude decay and phase shift compared to the solutions obtained with the remaining methods. These schemes cannot reflect the participation of high-frequency contributions in the response, because of their strong algorithmic dissipation. The results are consistent with the comparisons in Sect. 4. LMS2 and SS2 with $\rho_\infty = 0.0$ and most of the higher-order schemes have larger algorithmic dissipation than the other schemes. Therefore, when the contribution of high-frequencies needs to be considered in the solution, the second-order methods, especially LMS3, SS3, LMS4, SS4 with a large ρ_∞ , such as 0.6, are recommended.

5.4 Rotating shaft

Problem description Figure 21 shows the configuration of a rotating shaft. Its end R is connected to the ground by a revolute joint, whereas the other end T is connected to the

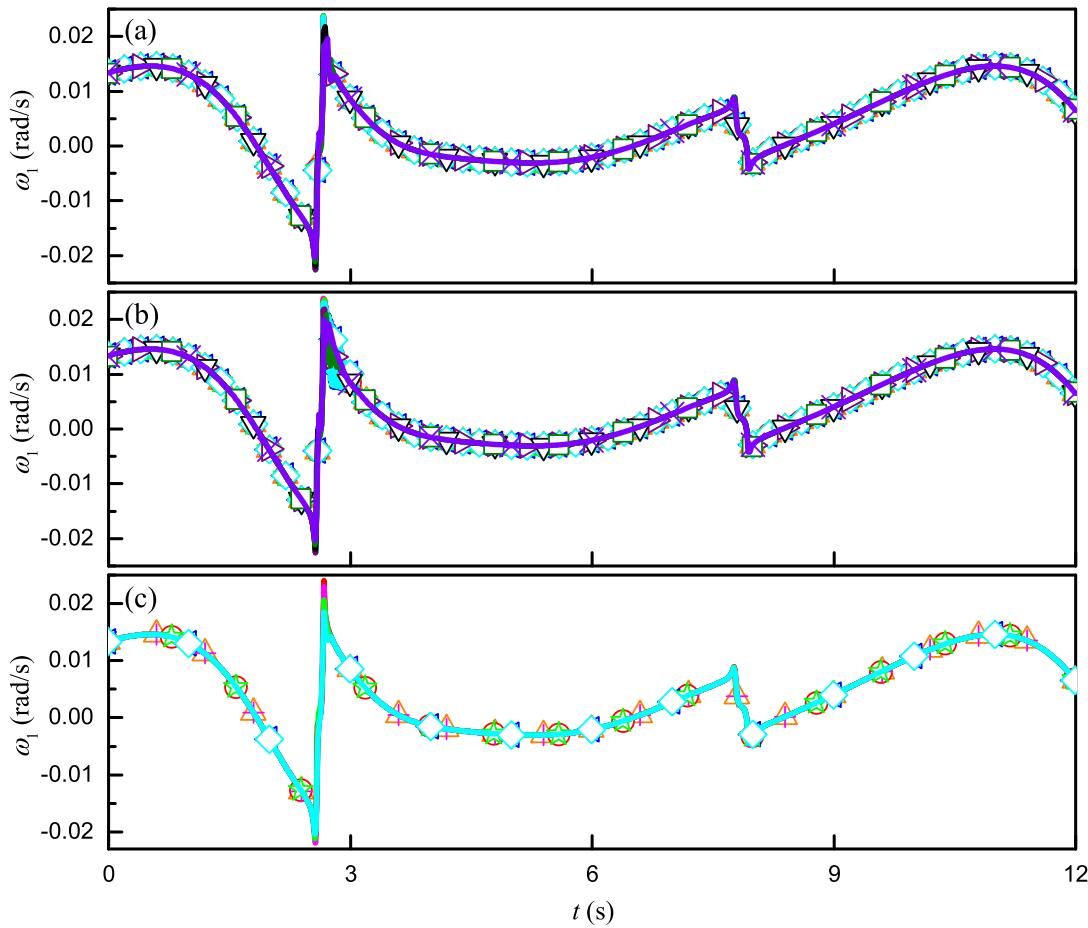


Fig. 15 Angular velocity ω_1 about the x axis at the tip of bar $B-C$ at point C within $[0, 12]$ s using $\Delta t/n = 4 \times 10^{-3}$ s **(a)** Second-order methods with $\rho_\infty = 0.0$; **(b)** Second-order methods with $\rho_\infty = 0.6$; **(c)** Higher-order methods with $\rho_\infty = 0.0$

ground by a cylindrical joint that allows rotation about and displacement along the shaft's axis. A rigid disk is attached to the shaft at the mid-span; its center of mass is offset from the reference axis of the shaft by $d = 0.05$ m in the z direction, thus inertially unbalancing the system. The shaft length is $L = 6$ m; its other properties are listed in Table 5. It is modelled with 16 three-node beam elements. The disk has mass $m_d = 70.573$ kg, radius $r_d = 0.24$ m, and thickness $t_d = 0.05$ m. Its inertial tensor with respect to the center of mass is $\text{diag}(2.0325, 1.0163, 1.0163) \text{ g} \cdot \text{m}^2$. The angular velocity of the revolute joint at the point R is prescribed as

$$\Omega = \begin{cases} \frac{1}{2}A_1\omega \left(1 - \cos \frac{\pi t}{T_1}\right), & 0 \leq t \leq T_1 \\ A_1\omega, & T_1 < t \leq T_2 \\ A_1\omega + \frac{1}{2}(A_2 - A_1)\omega \left(1 - \cos \frac{\pi(t - T_2)}{T_3 - T_2}\right), & T_2 < t \leq T_3 \\ A_2\omega, & t > T_3 \end{cases} \quad (25)$$

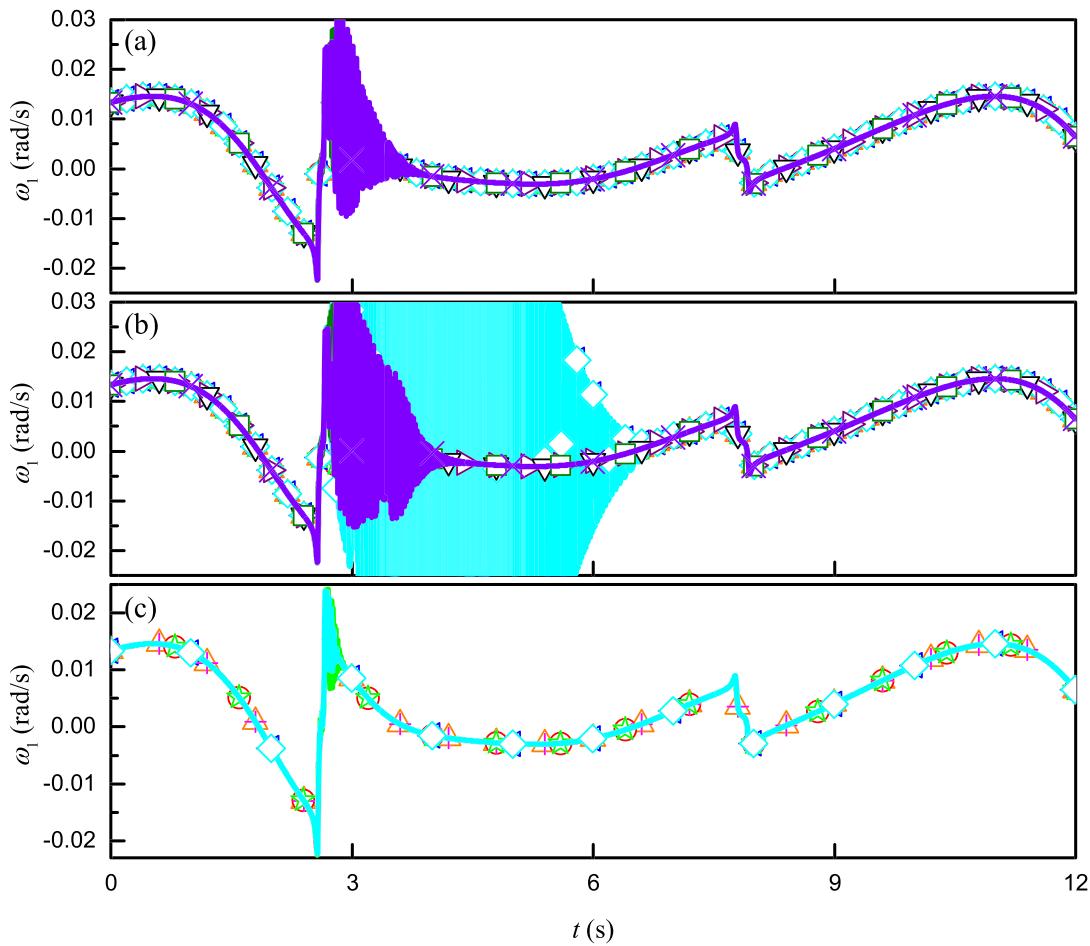
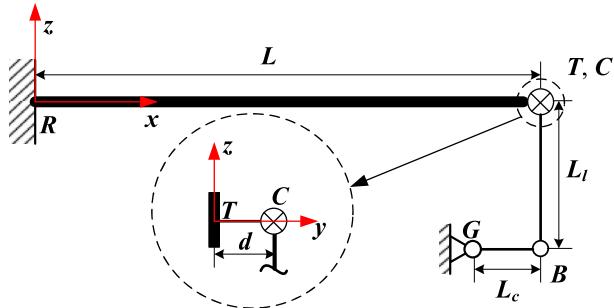


Fig. 16 Angular velocity ω_1 about the x axis at the tip of bar $B-C$ at point C within $[0, 12]$ s using $\Delta t/n = 1 \times 10^{-3}$ s **(a)** Second-order methods with $\rho_\infty = 0.0$; **(b)** Second-order methods with $\rho_\infty = 0.6$; **(c)** Higher-order methods with $\rho_\infty = 0.0$

Fig. 17 Crank actuated beam
(adapted from [4])



where $A_1 = 0.8$, $A_2 = 1.2$, $T_1 = 0.5$ s, $T_2 = 1$ s, $T_3 = 1.25$ s, $\omega = 60$ rad/s, i.e., it gently grows to an amplitude slightly below ω , pauses there for some time, and then gently grows again above ω , which corresponds roughly to a characteristic frequency of the system.

Numerical results The simulation was run in the interval $[-1, 2.5]$ s using $\Delta t/n = 10^{-3}$ s. Figures 22, 23 and 24 show, respectively, the displacement u_3 , velocity v_3 and acceleration a_3 along the z axis at the shaft's mid-span within $[0, 2.5]$ s. In this case, LMS4 with $\rho_\infty = 0.6$ failed to complete the solution. Excellent agreement can be observed between the results

Table 4 Inertia and stiffness properties of the crank, link and beam

	Beam	Link	Crank
Mass per unit span m (kg/m)	2.68	1.212	4.85
Moments of inertia per unit span J_1 (mg · m)	2255.33	87.30	1396.6
Moments of inertia per unit span J_2 (mg · m)	2233	43.65	698.3
Moments of inertia per unit span J_3 (mg · m)	22.33	43.65	698.3
Axial stiffness EA (MN)	73	33.02	132.1
Shearing stiffness GA_Y (MN)	5.025	10.81	43.22
Shearing stiffness GA_Z (MN)	23.40	10.81	43.22
Torsional stiffness GJ (N · m ²)	877.2	914.5	14630
Bending stiffness EJ_Y (N · m ²)	60830	1189	19020
Bending stiffness EJ_Z (N · m ²)	608.3	1189	19020

Table 5 Inertia and stiffness properties of the shaft

	Shaft
Mass per unit span m (kg/m)	11.64
Moments of inertia per unit span J_1 (mg · m)	26.34
Moments of inertia per unit span J_2 (mg · m)	13.17
Moments of inertia per unit span J_3 (mg · m)	13.17
Axial stiffness EA (MN)	313.4
Shearing stiffness GA_Y (MN)	60.5
Shearing stiffness GA_Z (MN)	60.5
Torsional stiffness GJ (N · m ²)	272.7
Bending stiffness EJ_Y (N · m ²)	354.5
Bending stiffness EJ_Z (N · m ²)	354.5

computed by the other employed methods, since the time step is sufficient to accurately describe the participating modes regardless of the algorithmic dissipation.

Convergence Based on this example, the convergence rates of the employed methods are investigated. Figures 25, 26 and 27 plot the relative errors of u_3 , v_3 and a_3 versus $\Delta t/n$ of the employed methods. The relative error RE is defined as

$$RE(x) = \sqrt{\frac{\sum_{j=1}^N (x(t_j) - x_j)^2}{\sum_{j=1}^N x(t_j)^2}} \quad (26)$$

where N is the number of total steps, $x(t_j)$ and x_j denote the exact and numerical solutions at time t_j , respectively. Since no exact solution can be obtained analytically for problems like this, it is replaced by the reference solution obtained using ESDIRK4(3)6L[2]SA₂ with a very small step size, $\Delta t = 10^{-5}$ s.

As can be seen, the second-order methods all exhibit a second-order convergence rate for displacement, velocity, and acceleration. The multi-step and equivalent single-step schemes

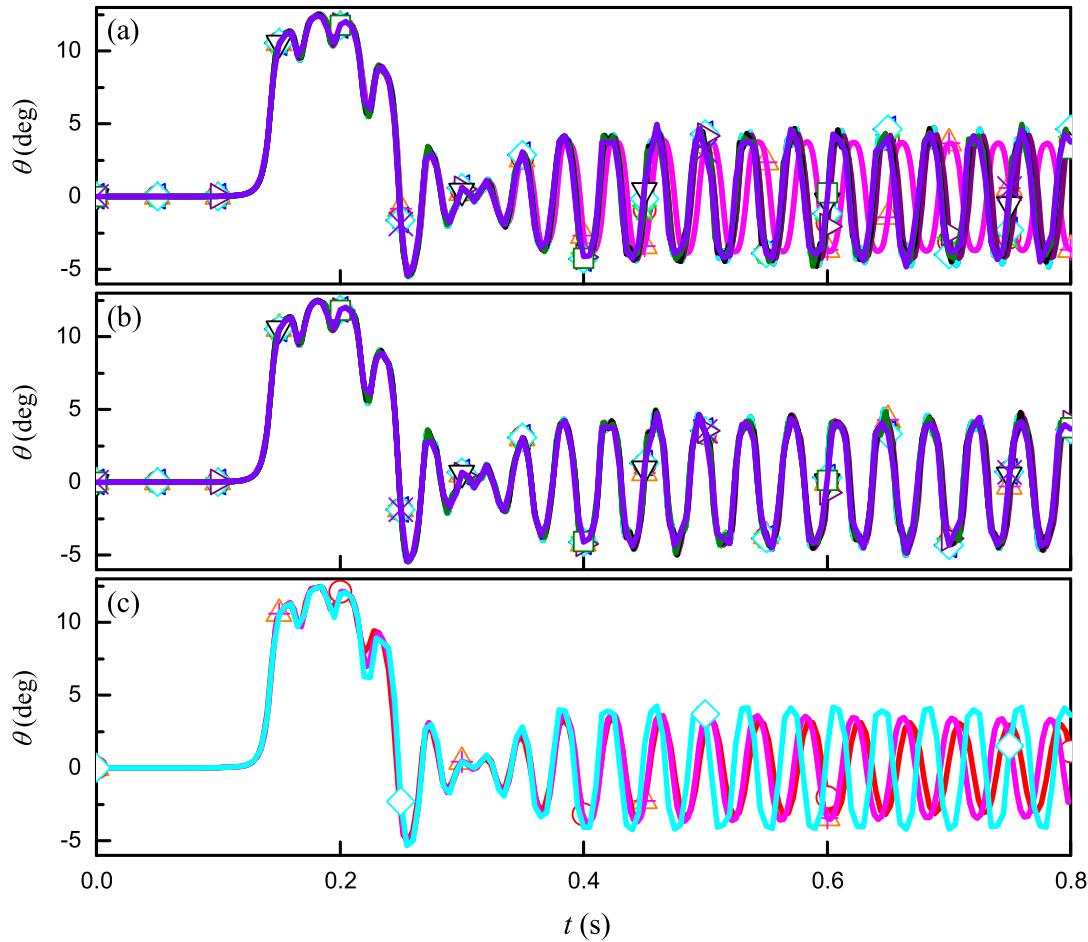


Fig. 18 Rotation angle θ about the x axis at the beam's mid-span within $[0, 0.8]$ s using $\Delta t/n = 10^{-3}$ s
(a) Second-order methods with $\rho_\infty = 0.0$; **(b)** Second-order methods with $\rho_\infty = 0.6$; **(c)** Higher-order methods with $\rho_\infty = 0.0$

have very similar relative errors. Accuracy improves with increasing number of steps or stages or with a large ρ_∞ . LMS2 and SS2 with $\rho_\infty = 0.0$ show larger errors than the other schemes. These conclusions are consistent with the analysis in Sect. 4.

Among the higher-order methods, the third-order MSSTH(3) and ESDIRK3(2)4L-[2]SA show a third-order convergence rate, and the third-order ESDIRK3(3)5L[2]SA shows a convergence rate exceeding the third-order. The fourth-order MSSTH(4) and ESDIRK4(3)6L[2]SA₂ show a convergence rate of about fourth-order when $\Delta t/n$ is close to 0.001 s, but when $\Delta t/n$ becomes smaller, their orders of convergence rate decrease, especially in a_3 . We think that this is because the reference solution, obtained using a very small step size, includes the contribution of a very large frequency range that contains many high frequencies not included in the numerical results. The treatment of these high-frequencies brings additional errors, which become more significant in the accelerations and in the cases where the error of the integrator in the frequency domain that it can retain is already quite small, such as the higher-order integrators with a small step size.

However, the fifth-order MSSTH(5) was never able to reach the fifth-order convergence rate. Regarding the reason, we think it may be related to the smoothness of the problem itself. Since the accuracy order is obtained from the Taylor expansion results, the higher-order

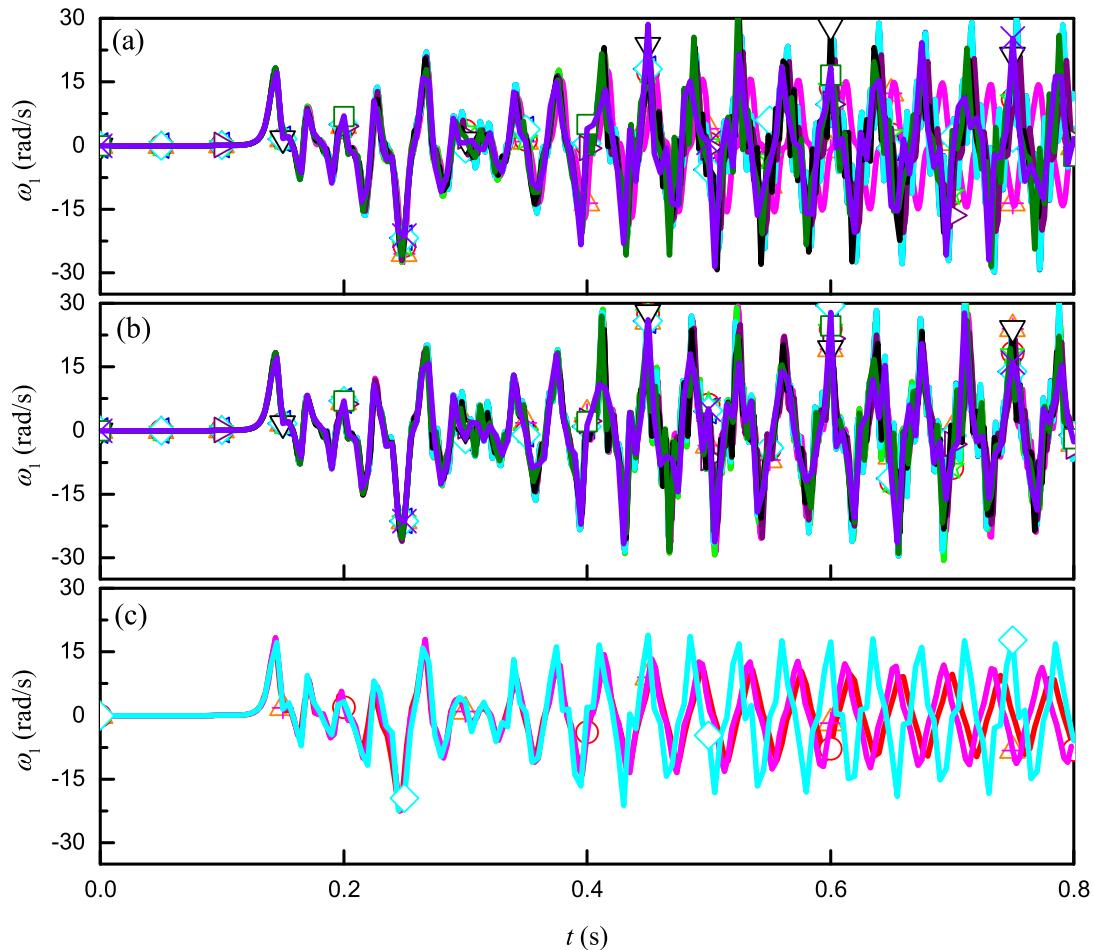


Fig. 19 Angular velocity ω_1 about the x axis at the beam's mid-span within $[0, 0.8]$ s using $\Delta t/n = 10^{-3}$ s
(a) Second-order methods with $\rho_\infty = 0.0$; **(b)** Second-order methods with $\rho_\infty = 0.6$; **(c)** Higher-order methods with $\rho_\infty = 0.0$

methods have higher requirements on the smoothness of the problems. If the smoothness does not meet the requirements, the order reduction may occur.

From the relative errors, ESDIRK3(3)5L[2]SA and ESDIRK4(3)6L[2]SA₂ are more accurate than other methods. The design of MSSTH(3, 4, 5) considers the minimization of the local truncation error, but these methods do not perform as expected. It indicates that a small local truncation error does not necessarily mean higher accuracy. The amplitude and period accuracy, as shown in Figs. 5–6, can better represent the accuracy of an integrator. Therefore, for high-accuracy purpose, ESDIRK3(3)5L[2]SA and ESDIRK4(3)6L[2]SA₂ are more recommended.

5.5 Average number of iterations

Since the same $\Delta t/n$ is used in the examples, the number of steps or sub-steps required for all methods is the same. However, as discussed in Sect. 2, the scheme used for prediction also plays an important role in computational efficiency. Therefore, the average number of iterations required for each step/sub-step in all examples solved in this section is listed in Fig. 28, to illustrate the cost of the employed methods. Two time step sizes are considered here for each example.

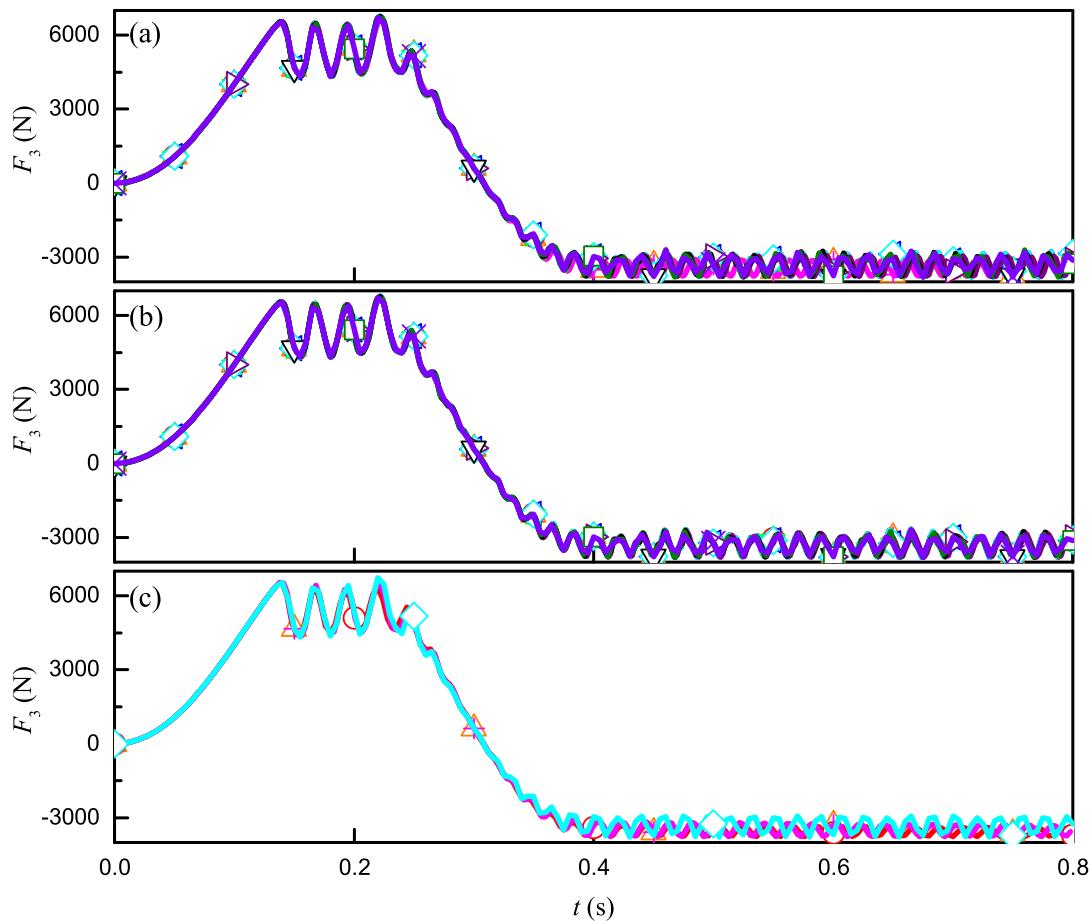
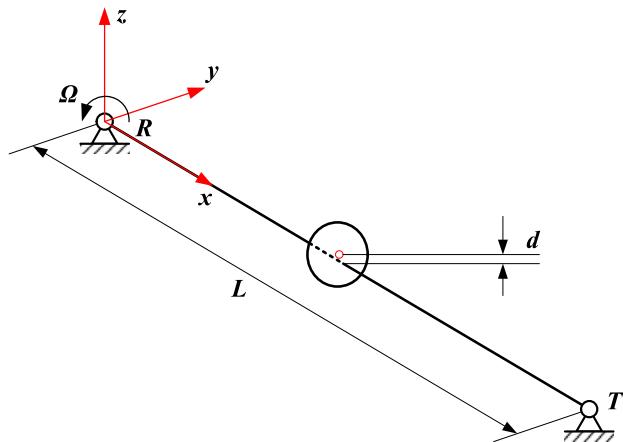


Fig. 20 Shear force F_3 at the beam's mid-span within $[0, 0.8]$ s using $\Delta t/n = 10^{-3}$ s **(a)** Second-order methods with $\rho_\infty = 0.0$; **(b)** Second-order methods with $\rho_\infty = 0.6$; **(c)** Higher-order methods with $\rho_\infty = 0.0$

Fig. 21 Rotating shaft with unbalanced disk (adapted from [4])



As shown in Fig. 28, the average number of iterations decreases with a smaller step size in all cases. The number of iterations spent by each integrator does not differ significantly. However, it can be observed that each single-step method always requires more iterations than the corresponding multi-step method. As discussed in Sect. 3, single-step methods use constant prediction, while multi-step methods use an explicit second-order scheme for

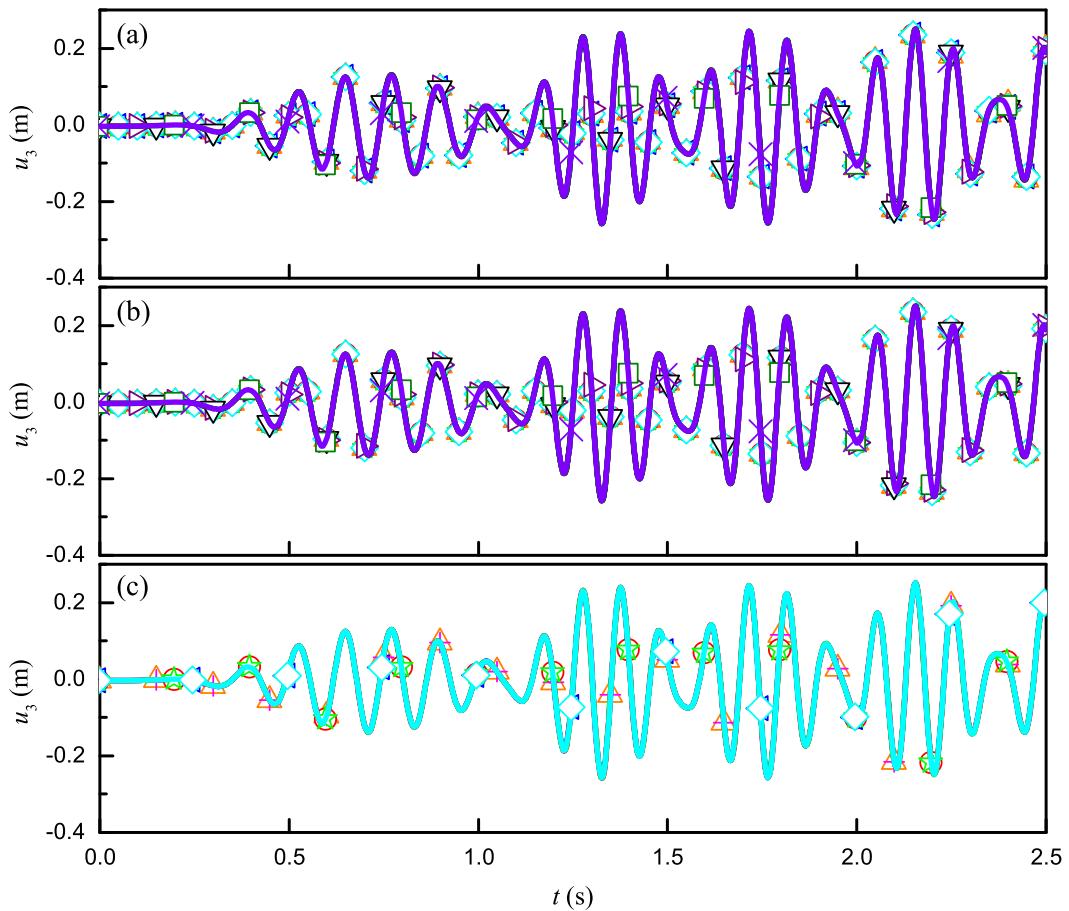


Fig. 22 Displacement u_3 at the shaft's mid-span within $[0, 2.5]$ s using $\Delta t/n = 10^{-3}$ s **(a)** Second-order methods with $\rho_\infty = 0.0$; **(b)** Second-order methods with $\rho_\infty = 0.6$; **(c)** Higher-order methods with $\rho_\infty = 0.0$

prediction. Therefore, this observation supports the conclusion that the use of an explicit second-order prediction scheme in these second-order methods helps to improve computational efficiency.

Besides, also the higher-order methods require more iterations than the remaining second-order methods in most cases. This may also be explained by the consideration that the accuracy of the prediction scheme is not close enough to that of the time integration scheme. The second-order multi-step and multi-stage methods usually require the least number of iterations of all methods. The data from Fig. 28 seem to support the consideration that a suitable prediction scheme, which must be explicit and have accuracy close to that of the integration scheme, is helpful in saving computational costs.

6 Conclusions

In this work, the performance of several representative implicit, A-stable time integration methods is discussed in view of their application to multibody system dynamics. The employed methods include linear two-, three-, and four-step methods, referred to as LMS2, LMS3, LMS4, their equivalent single-step methods, indicated as SS2, SS3, SS4, and several explicit first-stage, singly diagonally-implicit Runge–Kutta methods (ESDIRKs), indicated as Bathe, MSSTC(3, 4, 5), MSSTH(3, 4, 5), ESDIRK3(2)4L[2]SA, ESDIRK3(3)5L[2]SA,

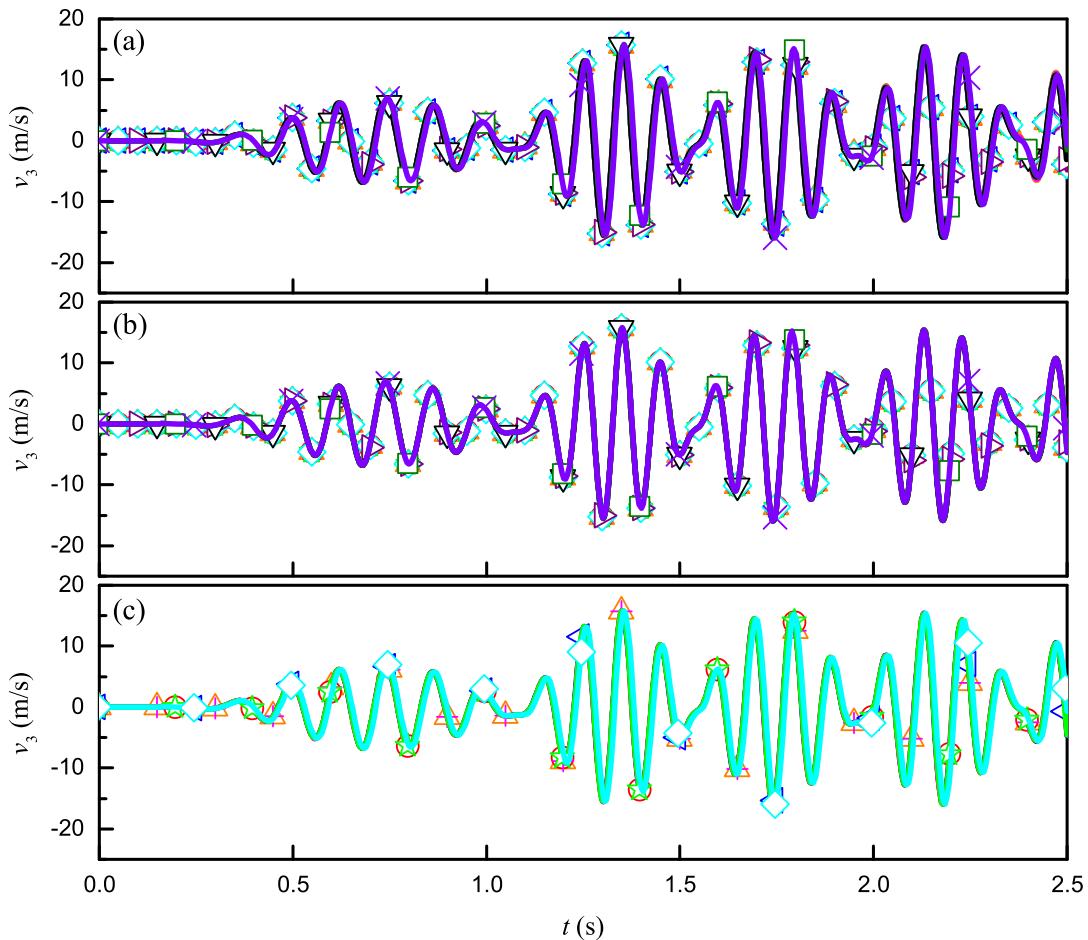


Fig. 23 Velocity v_3 at the shaft's mid-span within $[0, 2.5]$ s using $\Delta t/n = 10^{-3}$ s **(a)** Second-order methods with $\rho_\infty = 0.0$; **(b)** Second-order methods with $\rho_\infty = 0.6$; **(c)** Higher-order methods with $\rho_\infty = 0.0$

and ESDIRK4(3)6L[2]SA₂. These methods have been developed in [21, 35, 36], but the parameters of MSSTH(3, 4, 5) are modified here to satisfy the overall and stage order conditions. The formulations of the employed methods, and their implementation in the free general-purpose multibody solver MBDyn are presented.

In terms of properties, the linear multi-step, single-step, Bathe and MSSTC(3, 4, 5) methods have second-order accuracy and tunable algorithmic dissipation, whereas the other ESDIRKs can achieve higher-order accuracy. Several general conclusions can be drawn from the linear analysis and numerical experiments:

- with a suitable step size, all employed methods can predict accurate solutions.
- LMS3, SS3, LMS4 and SS4 with a large ρ_∞ , such as 0.6, show robust stability and good energy-conserving properties, making them suitable for long-term simulations and other cases where a large range of modes must be preserved, but these methods are not as good as others at filtering out high-frequency oscillations.
- LMS2 and SS2 with $\rho_\infty = 0.0$ (namely the second-order Backward Difference Formula) as well as most of the employed higher-order methods show a strong algorithmic dissipation even in the low-frequency range, so that their solutions are more likely to exhibit obvious amplitude decay and consequently a loss of accuracy at the large time steps.
- Bathe and the higher-order integrators with $\rho_\infty = 0.0$ can filter out high-frequency dynamics faster, so they are more useful for problems with high-frequency pollution.

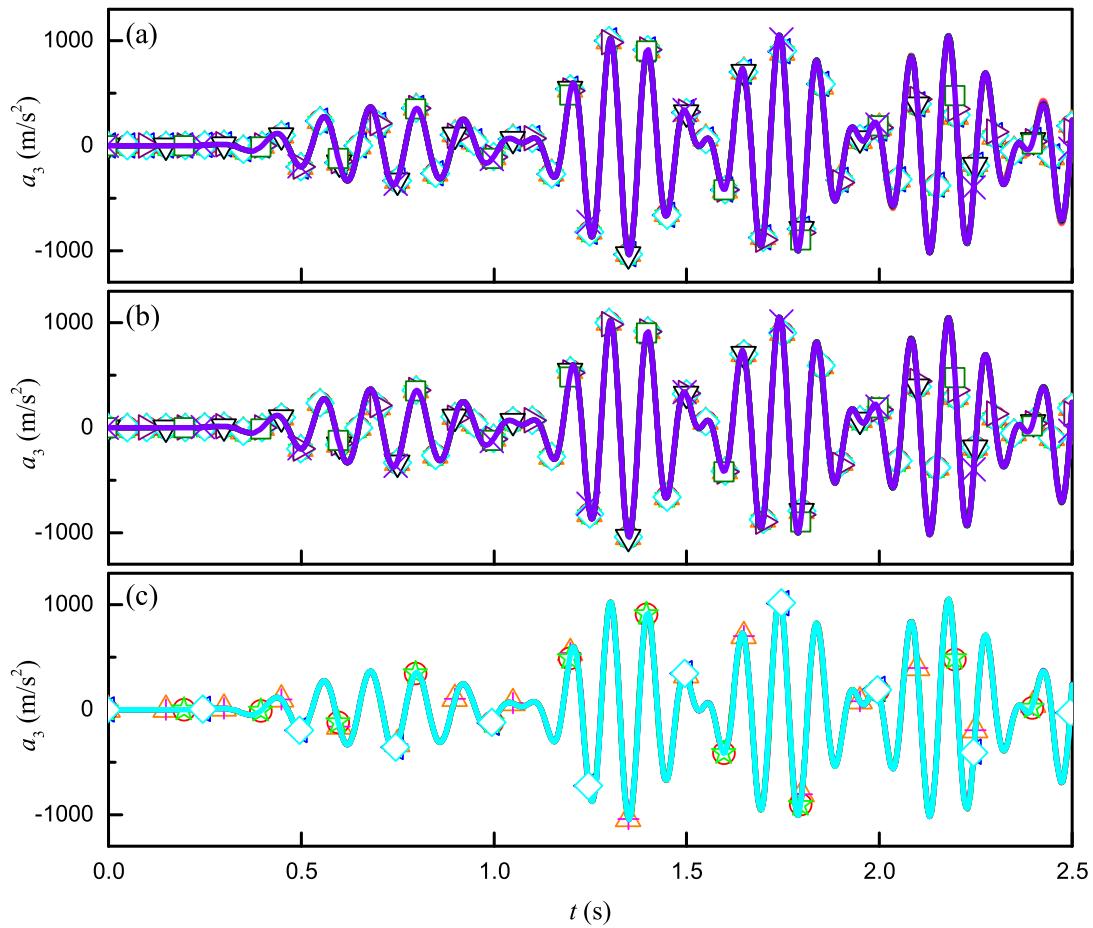


Fig. 24 Acceleration a_3 at the shaft's mid-span within $[0, 2.5]$ s using $\Delta t/n = 10^{-3}$ s **(a)** Second-order methods with $\rho_\infty = 0.0$; **(b)** Second-order methods with $\rho_\infty = 0.6$; **(c)** Higher-order methods with $\rho_\infty = 0.0$

- among the employed methods, ESDIRK3(3)5L[2]SA and ESDIRK4(3)6L[2]SA₂ have an obvious accuracy advantage over the other when the time step size is small enough to correctly integrate the participating dynamics, so they are recommended for high-accuracy purposes.
- the prediction scheme affects the number of iterations and thus the computational efficiency. It should be explicit, preferably if its order of accuracy is close to that of the time integration scheme.

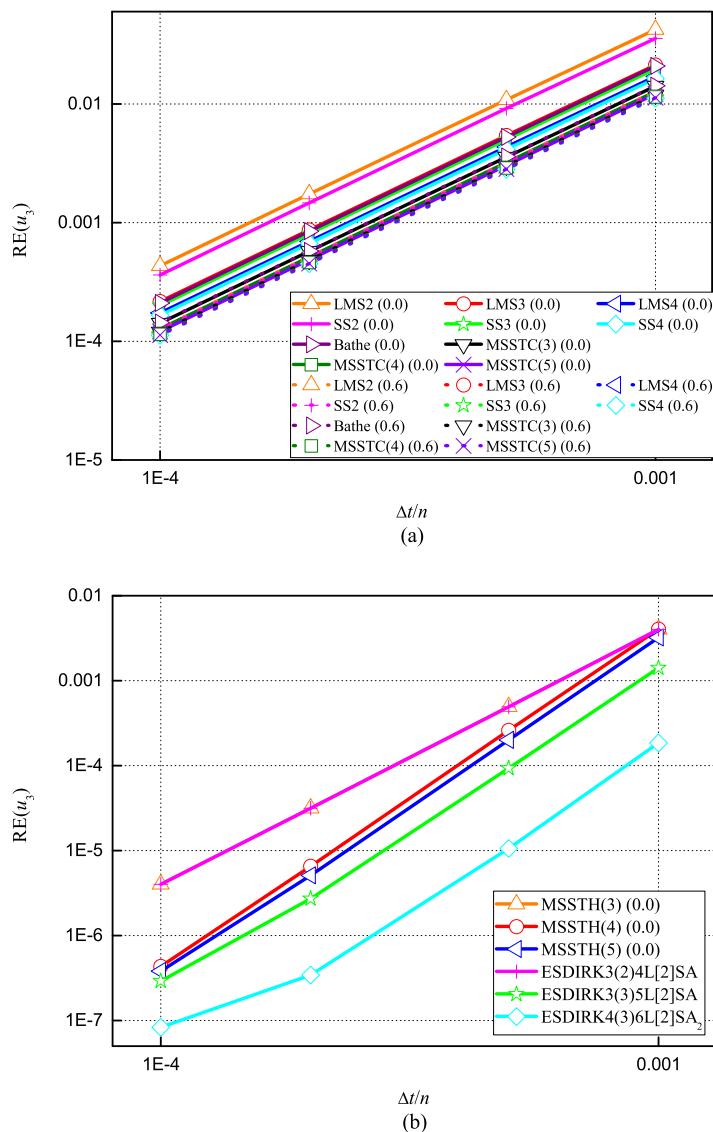
Appendix: Modified parameters of MSSTH(3, 4, 5)

Following the approach in [21], the local truncation error of an ESDIRK can be expressed as

$$\sigma = \sum_{i=1}^{\infty} (\Delta t)^i \sum_{j=1}^{a_i} \sigma_j^{(i)} F_j^{(i)} \quad (27)$$

where $F_j^{(i)}$ are elementary differentials, $a_i = \{1, 1, 2, 3, 9, 20\}$ for $i = \{1, 2, 3, 4, 5, 6\}$. The method is said to be p th-order accurate if $\sigma = O(\Delta t^{p+1})$, which requires that $\sigma_j^{(i)} = 0$ for

Fig. 25 Convergence rates of displacement u_3 at the shaft's mid-span for: (a) second-order methods; (b) higher-order methods (the value of ρ_∞ is placed in brackets in the legend)



all $j = 1, 2, \dots, a_i$ and $i = 1, 2, \dots, p$. The error of a p th-order method can be measured by

$$\|\boldsymbol{\sigma}^{(p+1)}\|_2 = \sqrt{\sum_{j=1}^{a_{p+1}} (\sigma_j^{(p+1)})^2} \quad (28)$$

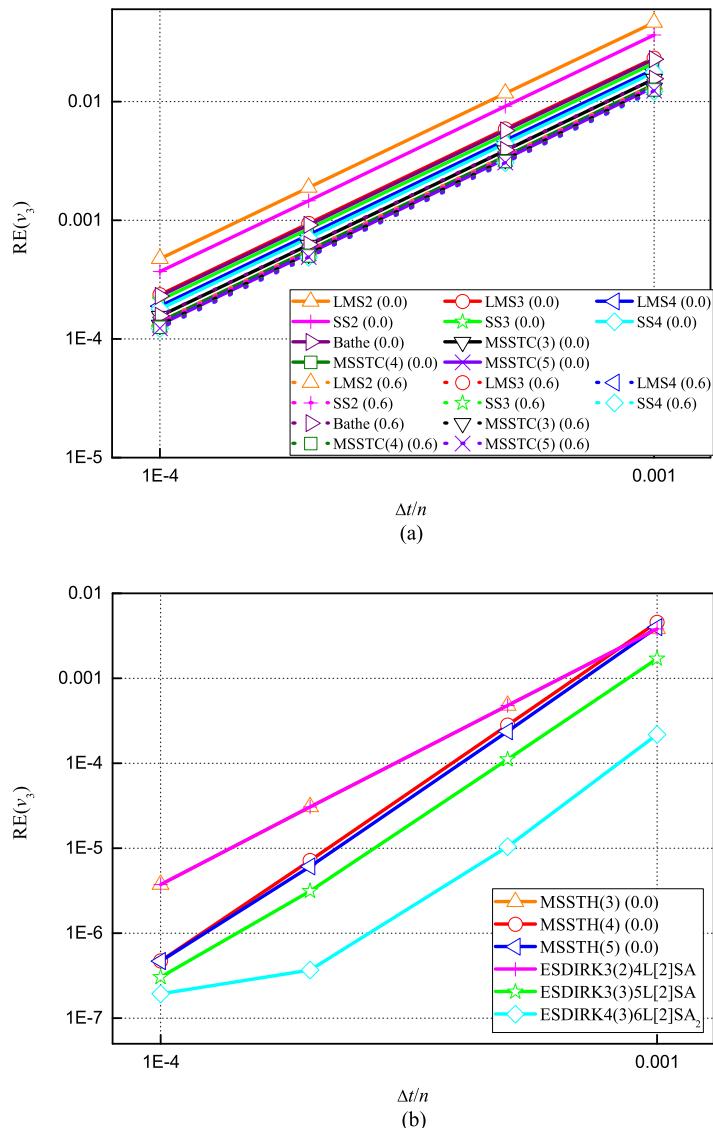
Here we use $\mathbf{A} = a_{ij}$, $\mathbf{b} = b_i$, $\mathbf{c} = c_i$, $\mathbf{C} = \text{diag}(\mathbf{c})$, $\mathbf{e} = [1, 1, 1, \dots, 1]^T$, and define

$$\mathbf{q}^{(k)} = \mathbf{AC}^{k-1}\mathbf{e} - \frac{1}{k}\mathbf{C}^k\mathbf{e}, \quad \mathbf{Q}^{(k)} = \text{diag}(\mathbf{q}^{(k)}) \quad (29)$$

the overall order conditions for up to sixth-order are expressed in Table 6. If $\mathbf{q}^{(1)} = \mathbf{q}^{(2)} = \dots = \mathbf{q}^{(\ell)} = \mathbf{0}$, the stage order of an ESDIRK is ℓ . According to the definition, one can check that MSSTC(n) has an overall order 2 and stage order 2.

MSSTH(n) is designed to provide high accuracy, so the conditions of stage order 2, overall order n , and a $\|\boldsymbol{\sigma}^{(n+1)}\|_2$ as small as possible are imposed. For MSSTH(3), $\mathbf{q}^{(1)} =$

Fig. 26 Convergence rates of velocity v_3 at the shaft's mid-span for: (a) second-order methods; (b) higher-order methods (the value of ρ_∞ is placed in brackets in the legend)



$\mathbf{q}^{(2)} = \mathbf{0}$, and $\|\boldsymbol{\sigma}^{(1)}\|_2 = \|\boldsymbol{\sigma}^{(2)}\|_2 = \|\boldsymbol{\sigma}^{(3)}\|_2 = 0$ yield

$$c_2 = 2\gamma \quad (30a)$$

$$a_{21} = \gamma \quad (30b)$$

$$a_{32} = \frac{c_3(c_3 - 2\gamma)}{4\gamma} \quad (30c)$$

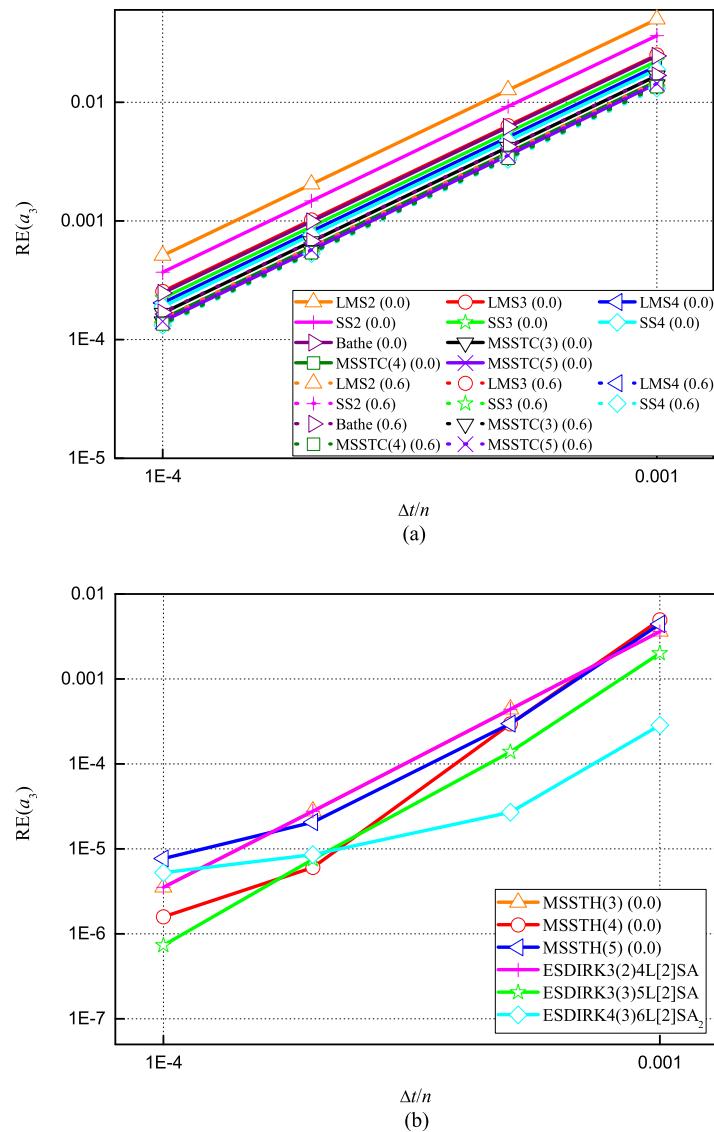
$$a_{31} = c_3 - \gamma - a_{32} \quad (30d)$$

$$b_2 = \frac{3c_3 + 6\gamma - 6c_3\gamma - 2}{12\gamma(c_3 - 2\gamma)} \quad (30e)$$

$$b_3 = \frac{6\gamma^2 - 6\gamma + 1}{3c_3(c_3 - 2\gamma)} \quad (30f)$$

$$b_1 = 1 - \gamma - b_2 - b_3 \quad (30g)$$

Fig. 27 Convergence rates of acceleration a_3 at the shaft's mid-span for: (a) second-order methods; (b) higher-order methods (the value of ρ_∞ is placed in brackets in the legend)



leaving two free parameters γ and c_3 . For a given ρ_∞ , γ has been determined in [35] to offer the prescribed degree of algorithmic dissipation, as listed in Table 7, and c_3 can be obtained by minimizing $\|\boldsymbol{\sigma}^{(4)}\|$ as

$$c_3 = \frac{24\gamma^2 - 20\gamma + 3}{24\gamma^2 - 24\gamma + 4} \quad (31)$$

For MSSTH(4), the order conditions $\mathbf{q}^{(1)} = \mathbf{q}^{(2)} = \mathbf{0}$, and $\|\boldsymbol{\sigma}^{(1)}\|_2 = \|\boldsymbol{\sigma}^{(2)}\|_2 = \|\boldsymbol{\sigma}^{(3)}\|_2 = \|\boldsymbol{\sigma}^{(4)}\|_2 = 0$ give

$$c_2 = 2\gamma \quad (32a)$$

$$a_{21} = \gamma \quad (32b)$$

$$a_{32} = \frac{c_3(c_3 - 2\gamma)}{4\gamma} \quad (32c)$$

$$a_{31} = c_3 - \gamma - a_{32} \quad (32d)$$

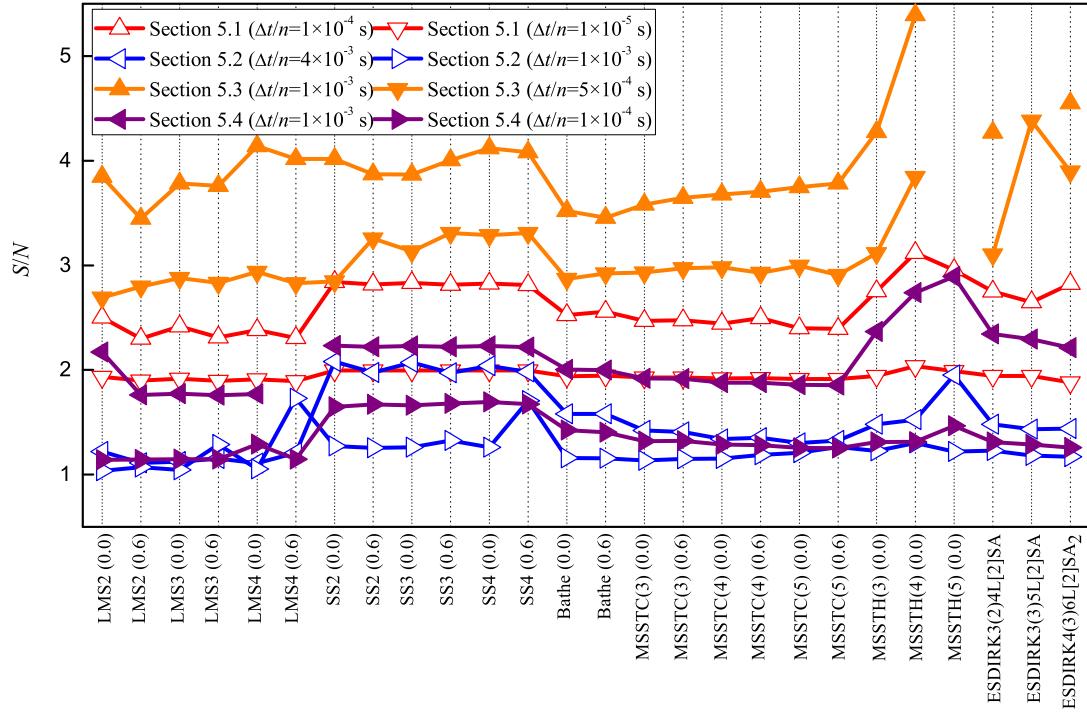


Fig. 28 Average number of iterations required in each step/sub-step S/N of the integrators in all examples (S is the total number of iterations, and N is the total number of steps/sub-steps)

$$a_{42} = \frac{c_4(c_4 - 2\gamma)\eta_1}{4\gamma\eta_2} \quad (32e)$$

$$a_{43} = \frac{c_4^2 - 4a_{42}\gamma - 2c_4\gamma}{2c_3} \quad (32f)$$

$$a_{41} = c_4 - \gamma - a_{42} - a_{43} \quad (32g)$$

$$b_2 = -\frac{12(c_3c_4 - c_3 - c_4 + 1)\gamma + 4c_3 + 4c_4 - 6c_3c_4 - 3}{24\gamma(c_3 - 2\gamma)(c_4 - 2\gamma)} \quad (32h)$$

$$b_3 = \frac{24(c_4 - 1)\gamma^2 + 4(5 - 6c_4)\gamma + 4c_4 - 3}{12c_3(c_4 - c_3)(c_3 - 2\gamma)} \quad (32i)$$

$$b_4 = -\frac{24(c_3 - 1)\gamma^2 + 4(5 - 6c_3)\gamma + 4c_3 - 3}{12c_4(c_4 - c_3)(c_4 - 2\gamma)} \quad (32j)$$

$$b_1 = 1 - \gamma - b_2 - b_3 - b_4 \quad (32k)$$

with

$$\eta_1 = 48(1 - c_4)\gamma^3 + 8(3c_3^2 - 6c_3 + 9c_4 - 5)\gamma^2 \quad (33a)$$

$$+ 6(-4c_3^2 + 6c_3 - 4c_4 + 1)\gamma + 4c_3^2 - 5c_3 + 2c_4$$

$$\eta_2 = 48(1 - c_3)\gamma^3 + 8(3c_3^2 + 3c_3 - 5)\gamma^2 \quad (33b)$$

$$+ 6(-4c_3^2 + 2c_3 + 1)\gamma + 4c_3^2 - 3c_3$$

Table 6 Order condition expressed using $\mathbf{q}^{(k)}$ for an ESDIRK

$\sigma_j^{(i)}$	Expression	$\sigma_j^{(i)}$	Expression
$\sigma_1^{(1)}$	$\mathbf{b}^T \mathbf{e} - 1$	$\sigma_1^{(2)}$	$\mathbf{b}^T \mathbf{C} \mathbf{e} - \frac{1}{2}$
$\sigma_1^{(3)}$	$\frac{1}{2} \mathbf{b}^T \mathbf{C}^2 \mathbf{e} - \frac{1}{6}$	$\sigma_2^{(3)}$	$\mathbf{b}^T \mathbf{q}^{(2)} + \sigma_1^{(3)}$
$\sigma_1^{(4)}$	$\frac{1}{6} \mathbf{b}^T \mathbf{C}^3 \mathbf{e} - \frac{1}{24}$	$\sigma_2^{(4)}$	$\mathbf{b}^T \mathbf{C} \mathbf{q}^{(2)} + 3\sigma_1^{(4)}$
$\sigma_3^{(4)}$	$\frac{1}{2} \mathbf{b}^T \mathbf{q}^{(3)} + \sigma_1^{(4)}$	$\sigma_4^{(4)}$	$\mathbf{b}^T \mathbf{A} \mathbf{q}^{(2)} + \sigma_3^{(4)}$
$\sigma_1^{(5)}$	$\frac{1}{24} \mathbf{b}^T \mathbf{C}^4 \mathbf{e} - \frac{1}{120}$	$\sigma_2^{(5)}$	$\frac{1}{2} \mathbf{b}^T \mathbf{C}^2 \mathbf{q}^{(2)} + 6\sigma_1^{(5)}$
$\sigma_3^{(5)}$	$\frac{1}{2} \mathbf{b}^T (\mathbf{Q}^{(2)} + \mathbf{C}^2) \mathbf{q}^{(2)} + 3\sigma_1^{(5)}$	$\sigma_4^{(5)}$	$\frac{1}{2} \mathbf{b}^T \mathbf{C} \mathbf{q}^{(3)} + 4\sigma_1^{(5)}$
$\sigma_5^{(5)}$	$\frac{1}{6} \mathbf{b}^T \mathbf{q}^{(4)} + \sigma_1^{(5)}$	$\sigma_6^{(5)}$	$\mathbf{b}^T \mathbf{C} \mathbf{A} \mathbf{q}^{(2)} + \sigma_4^{(5)}$
$\sigma_7^{(5)}$	$\mathbf{b}^T \mathbf{A} \mathbf{C} \mathbf{q}^{(2)} + 3\sigma_5^{(5)}$	$\sigma_8^{(5)}$	$\frac{1}{2} \mathbf{b}^T \mathbf{A} \mathbf{q}^{(3)} + \sigma_5^{(5)}$
$\sigma_9^{(5)}$	$\mathbf{b}^T \mathbf{A}^2 \mathbf{q}^{(2)} + \sigma_8^{(5)}$	$\sigma_1^{(6)}$	$\frac{1}{120} \mathbf{b}^T \mathbf{C}^5 \mathbf{e} - \frac{1}{720}$
$\sigma_2^{(6)}$	$\frac{1}{6} \mathbf{b}^T \mathbf{C}^3 \mathbf{q}^{(2)} + 10\sigma_1^{(6)}$	$\sigma_3^{(6)}$	$\frac{1}{2} \mathbf{b}^T \mathbf{C} (\mathbf{Q}^{(2)} + \mathbf{C}^2) \mathbf{q}^{(2)} + 15\sigma_1^{(6)}$
$\sigma_4^{(6)}$	$\frac{1}{4} \mathbf{b}^T \mathbf{C}^2 \mathbf{q}^{(3)} + 10\sigma_1^{(6)}$	$\sigma_5^{(6)}$	$\frac{1}{2} \mathbf{b}^T (\mathbf{Q}^{(3)} + \frac{1}{3} \mathbf{C}^3) \mathbf{q}^{(2)} + \sigma_4^{(6)}$
$\sigma_6^{(6)}$	$\frac{1}{6} \mathbf{b}^T \mathbf{C} \mathbf{q}^{(4)} + 5\sigma_1^{(6)}$	$\sigma_7^{(6)}$	$\frac{1}{24} \mathbf{b}^T \mathbf{q}^{(5)} + \sigma_1^{(6)}$
$\sigma_8^{(6)}$	$\frac{1}{2} \mathbf{b}^T \mathbf{C}^2 \mathbf{A} \mathbf{q}^{(2)} + \sigma_4^{(6)}$	$\sigma_9^{(6)}$	$\mathbf{b}^T \mathbf{Q}^{(2)} \mathbf{A} \mathbf{q}^{(2)} + \frac{1}{2} \mathbf{b}^T \mathbf{Q}^{(2)} \mathbf{q}^{(3)} + \frac{1}{6} \mathbf{b}^T \mathbf{C}^3 \mathbf{q}^{(2)} + \sigma_8^{(6)}$
$\sigma_{10}^{(6)}$	$\mathbf{b}^T \mathbf{C} \mathbf{A} \mathbf{C} \mathbf{q}^{(2)} + 3\sigma_6^{(6)}$	$\sigma_{11}^{(6)}$	$\frac{1}{2} \mathbf{b}^T \mathbf{A} \mathbf{C}^2 \mathbf{q}^{(2)} + 6\sigma_7^{(6)}$
$\sigma_{12}^{(6)}$	$\frac{1}{2} \mathbf{b}^T \mathbf{A} (\mathbf{Q}^{(2)} + \mathbf{C}^2) \mathbf{q}^{(2)} + 3\sigma_7^{(6)}$	$\sigma_{13}^{(6)}$	$\frac{1}{2} \mathbf{b}^T \mathbf{C} \mathbf{A} \mathbf{q}^{(3)} + \sigma_6^{(6)}$
$\sigma_{14}^{(6)}$	$\frac{1}{2} \mathbf{b}^T \mathbf{A} \mathbf{C} \mathbf{q}^{(3)} + 4\sigma_7^{(6)}$	$\sigma_{15}^{(6)}$	$\frac{1}{6} \mathbf{b}^T \mathbf{A} \mathbf{q}^{(4)} + \sigma_7^{(6)}$
$\sigma_{16}^{(6)}$	$\mathbf{b}^T \mathbf{A} \mathbf{C}^2 \mathbf{q}^{(2)} + \sigma_{13}^{(6)}$	$\sigma_{17}^{(6)}$	$\mathbf{b}^T \mathbf{A} \mathbf{C} \mathbf{A} \mathbf{q}^{(2)} + \sigma_{14}^{(6)}$
$\sigma_{18}^{(6)}$	$\mathbf{b}^T \mathbf{A}^2 \mathbf{C} \mathbf{q}^{(2)} + 3\sigma_{15}^{(6)}$	$\sigma_{19}^{(6)}$	$\frac{1}{2} \mathbf{b}^T \mathbf{A}^2 \mathbf{q}^{(3)} + \sigma_{15}^{(6)}$
$\sigma_{20}^{(6)}$	$\mathbf{b}^T \mathbf{A}^3 \mathbf{q}^{(2)} + \sigma_{19}^{(6)}$		

Then γ has been given to offer the tunable algorithmic dissipation, and c_3, c_4 are obtained by minimizing $\|\boldsymbol{\sigma}^{(5)}\|_2$ for several fixed ρ_∞ , as listed in Table 7.

For MSSTH(5), the order conditions impose

$$c_2 = 2\gamma \quad (34a)$$

$$a_{21} = \gamma \quad (34b)$$

$$a_{32} = \frac{c_3(c_3 - 2\gamma)}{4\gamma} \quad (34c)$$

$$a_{31} = c_3 - \gamma - a_{32} \quad (34d)$$

$$a_{42} = -\frac{c_4(c_4 - 2\gamma)\eta_1}{4\gamma(c_3 - 2\gamma)\eta_2} \quad (34e)$$

$$a_{43} = \frac{c_4^2 - 4a_{42}\gamma - 2c_4\gamma}{2c_3} \quad (34f)$$

$$a_{41} = c_4 - \gamma - a_{42} - a_{43} \quad (34g)$$

$$c_5 = \frac{\eta_3}{\eta_4} \quad (34h)$$

$$a_{53} = \frac{c_5(c_5 - c_3)(c_5 - 2\gamma)\eta_5}{c_3(c_4 - c_3)(c_3 - 2\gamma)\eta_6} \quad (34i)$$

$$a_{54} = \frac{c_5(c_5 - c_3)(c_5 - c_4)(c_5 - 2\gamma)\eta_7}{c_4(c_4 - c_3)(c_4 - 2\gamma)\eta_6} \quad (34j)$$

$$a_{52} = \frac{c_5^2 - 2a_{53}c_3 - 2a_{54}c_4 - 2c_5\gamma}{4\gamma} \quad (34k)$$

$$a_{51} = c_5 - \gamma - a_{52} - a_{53} - a_{54} \quad (34l)$$

$$b_3 = -\frac{\eta_8}{60c_3(c_4 - c_3)(c_5 - c_3)(c_3 - 2\gamma)} \quad (34m)$$

$$b_4 = \frac{\eta_9}{60c_4(c_4 - c_3)(c_5 - c_4)(c_4 - 2\gamma)} \quad (34n)$$

$$b_5 = -\frac{\eta_{10}}{60c_5(c_5 - c_3)(c_5 - c_4)(c_5 - 2\gamma)} \quad (34o)$$

$$b_2 = \frac{1 - 2\gamma - 2b_3c_3 - 2b_4c_4 - 2b_5c_5}{4\gamma} \quad (34p)$$

$$b_1 = 1 - \gamma - b_2 - b_3 - b_4 - b_5 \quad (34q)$$

with

$$\eta_1 = 240(1 - c_4)\gamma^4 + 40(3c_3^2 - 6c_3 + 12c_4 - 7)\gamma^3 \quad (35a)$$

$$+ 20(-9c_3^2 + 13c_3 - 12c_4 + 4)\gamma^2$$

$$+ (60c_3^2 - 70c_3 + 40c_4 - 6)\gamma - 5c_3^2 + 5c_3 - 2c_4$$

$$\eta_2 = 120(1 - c_3)\gamma^3 + 20(9c_3 - 7)\gamma^2 \quad (35b)$$

$$+ 20(2 - 3c_3)\gamma + 5c_3 - 3$$

$$\eta_3 = -720\gamma^5 + (360c_3 + 848)\gamma^4 \quad (35c)$$

$$+ (-336c_3 - 368)\gamma^3 + (120c_3 + 64)\gamma^2$$

$$+ (-18c_3 - 4)\gamma + c_3$$

$$\eta_4 = -480\gamma^5 + (240c_3 + 600)\gamma^4 \quad (35d)$$

$$+ (-240c_3 - 288)\gamma^3 + (96c_3 + 56)\gamma^2$$

Table 7 Parameters used in the modified MSSTH(n) for several fixed ρ_∞

ρ_∞	MSSTH(3): γ	MSSTH(4): γ	MSSTH(4): c_3
0.0	0.1804253064293983299659629	0.5728160624821350133117903	0.5590985754229417305152542
0.1	0.1786194582046580769940647	0.5483666449758298755412511	0.600293888698324121975147
0.2	0.1769458066182237054864146	0.5263864568423862744239727	0.6385228144891605953328610
0.3	0.1753855158428457572394876	0.5063301189707819505159136	0.6752454071331807752264900
0.4	0.1739236078771971283352116	0.4877974748123480863704060	0.7116626313535582440037008
0.5	0.1725479614220893076481644	0.4704805776216768320452388	0.7489373901316857945701066
0.6	0.1712486185906910429732619	0.4541307850365287057670116	0.7884370115210036155119526
0.7	0.1700172917724764587443786	0.4385361899021925080610629	0.8321495959968309360981758
0.8	0.1688470046791676892894429	0.4235037660671788772859259	0.8836585039419085905336449
0.9	0.167731825756886776530262	0.4088418661206993376389107	0.9508833135343227253337252
1.0	0.1666666666666666666666667	0.3943375672974065437870195	1.0534803411702867301702400
ρ_∞	MSSTH(4): c_4	MSSTH(5): γ	MSSTH(5): c_4
0.0	0.7414011664833654036144139	0.2780538411364499307154574	0.9673258605571696255864822
0.1	0.7584129875780372120885886	0.2741413060318684813410073	0.9085500184865173967097007
0.2	0.7731436659604612460228168	0.2704598867745817702967770	0.8510912674088796370241994
0.3	0.7860312064122737529814344	0.2669780439256505544243225	0.7951780419709296721109126
0.4	0.7972514819203143643377985	0.2636702317116055294121679	0.7409689864721386021173544
0.5	0.8067140747427041791439706	0.2605154166070549059952555	0.6885609787040850582329199
0.6	0.8139662529419134928687640	0.2574960298566747463056004	0.6379972790560722861741283
0.7	0.8179301032006714988753515	0.2545972081701334821524085	0.5892756783804685705163706
0.8	0.8162478946500242305006623	0.2518062311838496492022443	0.5423562209596147765111596
0.9	0.8036295352568830763217989	0.2491120965296297062874231	0.4971683414199104533715001
1.0	0.7689305362617052663765094	0.2465051931428201559270974	0.4536172576687555468843982

$$+ (-16c_3 - 4)\gamma + c_3$$

$$\eta_5 = 120(-c_4^2 + 2c_4 - c_3 - c_5 + c_3c_5)\gamma^3 \quad (35e)$$

$$+ 20(9c_4^2 - 15c_4 + 8c_3 + 7c_5 - 9c_3c_5)\gamma^2$$

$$+ 10(-6c_4^2 + 9c_4 - 5c_3 - 4c_5 + 6c_3c_5)\gamma$$

$$+ 5c_4^2 - 7c_4 + 4c_3 + 3c_5 - 5c_3c_5$$

$$\eta_6 = 120(c_3 + c_4 - c_3c_4 - 1)\gamma^2 \quad (35f)$$

$$+ 10(12c_3c_4 - 10c_4 - 10c_3 + 9)\gamma$$

$$+ 15c_3 + 15c_4 - 20c_3c_4 - 12$$

$$\eta_7 = 120(1 - c_3)\gamma^3 + 20(9c_3 - 7)\gamma^2 \quad (35g)$$

$$+ 20(2 - 3c_3)\gamma + 5c_3 - 3$$

$$\eta_8 = 120(c_4 + c_5 - c_4c_5 - 1)\gamma^2 \quad (35h)$$

$$+ 10(12c_4c_5 - 10c_5 - 10c_4 + 9)\gamma$$

$$+ 15c_4 + 15c_5 - 20c_4c_5 - 12 \\ \eta_9 = 120(c_3 + c_5 - c_3c_5 - 1)\gamma^2 \quad (35i)$$

$$+ 10(12c_3c_5 - 10c_5 - 10c_3 + 9)\gamma \\ + 15c_3 + 15c_5 - 20c_3c_5 - 12 \\ \eta_{10} = 120(c_3 + c_4 - c_3c_4 - 1)\gamma^2 \quad (35j)$$

$$+ 10(12c_3c_4 - 10c_4 - 10c_3 + 9)\gamma \\ + 15c_3 + 15c_4 - 20c_3c_4 - 12$$

leaving γ , c_3 and c_4 free. Here we assume $c_3 = 0.1$, and then c_4 is obtained by minimizing $\|\sigma^{(6)}\|_2$. The values used for γ and c_4 are listed in Table 7.

Acknowledgements The first and second authors acknowledge the financial support by the China Scholarship Council.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Alexander, R.: Diagonally implicit Runge–Kutta methods for stiff ODEs. *SIAM J. Numer. Anal.* **14**(6), 1006–1021 (1977)
2. Arnold, M., Brüls, O.: Convergence of the generalized- α scheme for constrained mechanical systems. *Multibody Syst. Dyn.* **18**(2), 185–202 (2007)
3. Ascher, U.M., Petzold, L.R.: Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, vol. 61. SIAM, Philadelphia (1998)
4. Bauchau, O.A., Betsch, P., Cardona, A., Gerstmayr, J., Jonker, B., Maserati, P., Sonnevile, V.: Validation of flexible multibody dynamics beam formulations using benchmark problems. *Multibody Syst. Dyn.* **37**(1), 29–48 (2016)
5. Belytschko, T., Schoeberle, D.: On the unconditional stability of an implicit algorithm for nonlinear structural dynamics. *J. Appl. Mech.* (1975). <https://doi.org/10.1115/1.3423721>
6. Boom, P.D., Zingg, D.W.: Optimization of high-order diagonally-implicit Runge–Kutta methods. *J. Comput. Phys.* **371**, 168–191 (2018)
7. Brenan, K.E., Campbell, S.L.V., Petzold, L.R.: Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. North-Holland, New York (1989)
8. Brüls, O., Arnold, M.: The generalized- α scheme as a linear multistep integrator: toward a general mechatronic simulator. *J. Comput. Nonlinear Dyn.* **3**(4), 041007 (2008). <https://doi.org/10.1115/1.2960475>
9. Butcher, J.C.: Implicit Runge–Kutta processes. *Math. Comput.* **18**(85), 50–64 (1964)
10. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations. Wiley, New York (2016)
11. Chung, J., Hulbert, G.: A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- α method. *J. Appl. Mech.* **60**, 371–375 (1993)

12. Dahlquist, G.G.: A special stability problem for linear multistep methods. *BIT Numer. Math.* **3**(1), 27–43 (1963)
13. de Alwis, T.: Central difference formula in numerical analysis. *PRIMUS, Probl. Resour. Issues Math. Undergrad. Stud.* **2**(2), 165–172 (1992)
14. Faltinsen, S., Marthinsen, A., Munthe-Kaas, H.Z.: Multistep methods integrating ordinary differential equations on manifolds. *Appl. Numer. Math.* **39**(3), 349–365 (2001). [https://doi.org/10.1016/S0168-9274\(01\)00103-9](https://doi.org/10.1016/S0168-9274(01)00103-9)
15. Ghiringhelli, G.L., Maserati, P., Mantegazza, P.: Multibody implementation of finite volume C^0 beams. *AIAA J.* **38**(1), 131–138 (2000)
16. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations, II, Stiff and Differential-Algebraic Problems. Springer, Berlin (1991)
17. Hilber, H.M., Hughes, T.J., Taylor, R.L.: Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthq. Eng. Struct. Dyn.* **5**(3), 283–292 (1977)
18. Hoff, C., Pahl, P.: Development of an implicit method with numerical dissipation from a generalized single-step algorithm for structural dynamics. *Comput. Methods Appl. Mech. Eng.* **67**(3), 367–385 (1988)
19. Houboldt, J.C.: A recurrence matrix solution for the dynamic response of elastic aircraft. *J. Aeronaut. Sci.* **17**(9), 540–550 (1950)
20. Jansen, K.E., Whiting, C.H., Hulbert, G.M.: A generalized- α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. *Comput. Methods Appl. Mech. Eng.* **190**(3–4), 305–319 (2000). [https://doi.org/10.1016/S0045-7825\(00\)00203-6](https://doi.org/10.1016/S0045-7825(00)00203-6)
21. Kennedy, C.A., Carpenter, M.H.: Diagonally implicit Runge–Kutta methods for stiff ODEs. *Appl. Numer. Math.* **146**, 221–244 (2019)
22. Köbis, M., Arnold, M.: Convergence of generalized- α time integration for nonlinear systems with stiff potential forces. *Multibody Syst. Dyn.* **37**(1), 107–125 (2016)
23. Maserati, P., Lanz, M., Mantegazza, P.: Multistep integration of ordinary, stiff and differential-algebraic problems for multibody dinamics applications. In: XVI Congresso Nazionale AIDAA, pp. 1–10 (2001)
24. Maserati, P., Morandini, M., Mantegazza, P.: An efficient formulation for general-purpose multi-body/multiphysics analysis. *J. Comput. Nonlinear Dyn.* **9**(4), 041001 (2014)
25. Newmark, N.M.: A method of computation for structural dynamics. *J. Eng. Mech. Div.* **85**(3), 67–94 (1959)
26. Noh, G., Bathe, K.J.: The Bathe time integration method with controllable spectral radius: the ρ_∞ -Bathe method. *Comput. Struct.* **212**, 299–310 (2019)
27. Owren, B., Simonsen, H.H.: Alternative integration methods for problems in structural dynamics. *Comput. Methods Appl. Mech. Eng.* **122**(1–2), 1–10 (1995). [https://doi.org/10.1016/0045-7825\(94\)00717-2](https://doi.org/10.1016/0045-7825(94)00717-2)
28. Park, K.: An improved stiffly stable method for direct integration of nonlinear structural dynamic equations. *J. Appl. Mech.* **42**(3), 464–470 (1975)
29. Schiehlen, W., et al.: Multibody Systems Handbook, vol. 6. Springer, Berlin (1990)
30. Tamma, K.K., Har, J., Zhou, X., Shimada, M., Hoitink, A.: An overview and recent advances in vector and scalar formalisms: space/time discretizations in computational dynamics – a unified approach. *Arch. Comput. Methods Eng.* **18**(2), 119–283 (2011)
31. Wilson, E.L.: A computer program for the dynamic stress analysis of underground structures. Tech. rep., California Univ Berkeley Structural Engineering Lab (1968)
32. Wood, W., Bossak, M., Zienkiewicz, O.: An alpha modification of Newmark's method. *Int. J. Numer. Methods Eng.* **15**(10), 1562–1566 (1980)
33. Xie, Y., Steven, G.P.: Instability, chaos, and growth and decay of energy of time-stepping schemes for non-linear dynamic equations. *Commun. Numer. Methods Eng.* **10**(5), 393–401 (1994)
34. Zhang, J.: A-stable two-step time integration methods with controllable numerical dissipation for structural dynamics. *Int. J. Numer. Methods Eng.* **121**, 54–92 (2020)
35. Zhang, H., Zhang, R., Xing, Y., Maserati, P.: On the optimization of n -sub-step composite time integration methods. *Nonlinear Dyn.* **102**(3), 1939–1962 (2020)
36. Zhang, H., Zhang, R., Maserati, P.: Improved second-order unconditionally stable schemes of linear multi-step and equivalent single-step integration methods. *Comput. Mech.* **67**(1), 289–313 (2021). <https://doi.org/10.1007/s00466-020-01933-y>
37. Zhou, X., Tamma, K.K.: Design, analysis, and synthesis of generalized single step single solve and optimal algorithms for structural dynamics. *Int. J. Numer. Methods Eng.* **59**(5), 597–668 (2004)

Pierangelo Masarati

Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano

via La Masa 34, 20156 Milano, Italy

Tel.: ++39 02 2399 8309

Fax: ++39 02 2399 8334

E-mail: masarati@aero.polimi.it

Web: <http://www.aero.polimi.it/~mbdyn/>

Web: <http://www.aero.polimi.it/~masarati/>

Web: <http://www.aero.polimi.it/~masarati/MBDyn-input/manual/index.html>