

# Coordinate transformation techniques for efficient model reduction in flexible multibody dynamics

G. Heirman<sup>1</sup>, O. Brüls<sup>2</sup>, P. Sas<sup>1</sup>, W. Desmet<sup>1</sup>

<sup>1</sup> Katholieke Universiteit Leuven, Department of Mechanical Engineering, Celestijnenlaan 300 B, B-3001 Heverlee, Belgium

<sup>2</sup> University of Liège, LTAS-Vibrations et Identification des Structures, Chemin des Chevreuils 1, B52/3, B-4000 Liège, Belgium

e-mail: [gert.heirman@mech.kuleuven.be](mailto:gert.heirman@mech.kuleuven.be)

## Abstract

Computational efficiency is important for all numerical simulation tools. For real-time and faster-than-real-time applications, which rely on a strong interaction between simulation results and other subsystems, it is vital. This paper proposes a theoretical framework for coordinate transformations to recast the differential-algebraic system equations of a flexible mechanism into a simpler set of equations, which is cheaper to solve. Desirable properties of the coordinate transformation to minimize the computational burden of the simulation are discussed, as well as some assumptions that can be made for further simplification. A methodology to make practical use of coordinate transformation techniques to speed up simulation speed for real-time and faster-than-real-time applications is presented.

## 1 Introduction

Computer simulations are of ever growing importance in engineering. They are used for design purposes, but for an important number of applications, simulation is the core of the application itself. In these applications computer simulation closely interacts with other subsystems of the application. Examples are Hardware-in-the-Loop, Human-in-the-Loop and Software-in-the-Loop. In these applications, the interaction of a subsystem (a piece of hardware, a human or software) and the remainder of the envisaged system needs to be evaluated. For economic or safety reasons and/or lack of time it is impossible to let it interact with the actual remainder of the envisaged system. Instead the remainder of the system is modelled. This model is fed by the output of the hardware/human/software and the output of the model is fed back to the hardware/human/software. Obviously this requires fast generation of accurate simulation results.

Other applications like Model-based Predictive Control require even faster-than-real-time simulation results. In Model-based Predictive Control one wishes to evaluate the effect of different control inputs. Based on the simulation results, the control input with the best predicted results will be selected and executed. The simulation of the effect of the different control inputs should thus be performed in a time span much shorter than the simulated process, otherwise the selected optimal control input will become obsolete before it can be executed. Another example is model-based condition monitoring, an application in which the correct functioning of a machine is checked by comparing measurements with simulation results of a model which is fed by the same inputs. Again, this requires a fast calculating, accurate computer simulation of the monitored machine.

In the field of (flexible) mechanisms, the current simulation techniques are not powerful enough for this kind of real-time or even faster-than-real-time applications, unless if the models are heavily simplified, rendering their simulation results inaccurate. A first obvious solution to speed up calculations is through the use of

better hardware. This solution is however not sufficient, as the average simulation time for typical engineering problems doesn't decrease as hardware improves. Apparently better hardware only tempts engineers to simulate more complex system models. Other improvements are required.

It is important to note that most real-time and faster-than-real-time applications allow for an expensive preparation phase, as long as the computational load during the actual simulation is kept low. This opportunity is often not fully exploited in current techniques.

This paper provides a theoretical frame to speed up simulations of flexible mechanisms by projecting the initial set of equations upon a new coordinate system. In a first chapter problems for fast simulation of flexible mechanisms will be highlighted. The next chapter treats the actual coordinate transformation and describes how this transformation should be defined in order to maximally benefit from it. Then a methodology is proposed how this coordinate transformation should be used to fit for the purpose of real-time or faster-than-real-time calculation. Finally, a modal representation of the flexibility of the mechanism is discussed, aiming to maximally reduce the computational burden of the computer simulation.

## 2 Problem Description

### 2.1 Equations of motion

Many choices of coordinates are possible for the formulation of a (flexible) multibody system [5]: Cartesian, natural and relative coordinates. The choice is a trade-off between computational efficiency and the generic character of the approach. The optimal choice of the coordinates used, depends on the modeled system and the envisaged application. Irrespective of the used type of coordinates, the equations of motion of a (flexible) multibody system can be written as a second order differential-algebraic equation (DAE):

$$\begin{aligned} M^{qq}(q) \ddot{q} + h^q(q, \dot{q}) + \mathcal{V}_{,q} + \Phi_{,q}^T \lambda &= g^q & (1) \\ \Phi(q) &= 0 & (2) \end{aligned}$$

In these equations:

- $q$  is a vector of the  $n$  generalized coordinates, the value of  $q$  defines the *configuration* of the system
- $M^{qq}(q)$  is the configuration dependent mass matrix
- $\mathcal{V}_{,q}$  is the gradient of the potential energy. Only potential energy due to structural deformation is considered in this term. The generalized forces due to other potential energy sources (e.g. gravity) will be taken into account by the source term  $g^q$ . As the potential energy can be shifted by an arbitrary value,  $\mathcal{V} = 0$  is chosen to correspond to an undeformed configuration.
- $g^q$  denotes the generalized forces due to external loads on a component
- $\Phi(q) = 0$  expresses the  $m$  kinematic holonomic constraints<sup>1</sup>. Its gradient  $\Phi_{,q}^T$  is assumed to be of full rank for all configurations  $q$ .
- $\lambda$  is a vector of  $m$  Lagrange multipliers to take the constraints into account.
- $\Phi_{,q}^T \lambda$  represents the reaction forces and moments enforcing the constraints<sup>2</sup>.

<sup>1</sup>Non-holonomic constraints are not considered in this work.

<sup>2</sup>A gradient of an entity  $A$  with respect to a vector  $b$  will be denoted as  $A_{,b}$ .

- $h^q$  gathers the centrifugal and Coriolis inertia forces which are quadratic in  $\dot{q}$ . Using the index summation convention, we have

$$(h^q)_i = (\Gamma^{qqq})_{ijk} \dot{q}_j \dot{q}_k \tag{3}$$

where  $(\Gamma^{qqq})_{ijk}$  is the Christoffel symbol of the first kind:

$$(\Gamma^{qqq})_{ijk} = \frac{1}{2} \left( \frac{\partial(M^{qq})_{ij}}{\partial q_k} + \frac{\partial(M^{qq})_{ik}}{\partial q_j} - \frac{\partial(M^{qq})_{jk}}{\partial q_i} \right) \tag{4}$$

The system will be assumed to have  $s$  rigid body degrees of freedom, it is modeled to have  $n - m$  intrinsic system degrees of freedom. For a flexible mechanism we have the inequality

$$s \leq n - m \tag{5}$$

This is an equality in case of a rigid multibody system.

If the flexibility of the system needs to be represented accurately, a lot of generalized coordinates are needed [5], thus  $n - m$  will be considerably larger than  $s$ . The flexibility of each component can also be represented by means of a modal representation, limiting the number of needed generalized coordinates  $q$  to obtain reasonable accuracy. Modal reduction requires inevitably some kind of linearization, therefore phenomena such as geometric stiffening and/or non-linear material behaviour will become tedious to be described in a modal description [6]. This work doesn't focus on geometric stiffening of components nor on non-linear material behavior. However, the proposed methodology could be extended to describe these phenomena if linearizations around deformed states are performed.

### 2.2 The different subspaces

The  $n$  coordinates in vector  $q$  form an  $\mathbb{R}^n$  space. The  $m$  constraint equations (2) limit the possible values of  $q$  to a subspace of dimension  $n - m$ , the *flexible configuration space*  $\widehat{\Omega}_q^{ft}$  [1]. The *rigid configuration space*  $\Omega_q^{rt} \subseteq \widehat{\Omega}_q^{ft}$ , with intrinsic dimension  $s$ , is defined as the set of kinematically admissible undeformed configurations:

$$\Omega_q^{rt} = \left\{ q \in \widehat{\Omega}_q^{ft} \mid \mathcal{V}(q) = 0 \right\} \tag{6}$$

The *deformed configuration space*  $\Omega_q^{dt}$  are all the kinematically admissible deformed configurations  $q$ . Thus  $\Omega_q^{dt} \equiv \widehat{\Omega}_q^{ft} \setminus \Omega_q^{rt}$ .

### 2.3 Difficulties for fast simulation

For most applications, the values of the degrees of freedom  $q$ , and their time derivatives  $\dot{q}$ ,  $\ddot{q}$ , as well as the reaction forces  $\Phi_{,q}^T \lambda$  are of interest within a certain time frame, the *simulated time*. Once these values are known, material stresses and strains can be obtained as a postprocessing result. Assuming the values of the degrees of freedom and the Lagrange multipliers to be known at time  $t_i$ , the system equations need to be integrated over the time step  $h_i$  to obtain the values of the degrees of freedom and the Lagrange multipliers at the next discretized time point  $t_{i+1}$ . To converge to a solution for the degrees of freedom and the Lagrange multipliers at time  $t_{i+1}$ , which satisfies (1) and (2) up to a user-specified tolerance, multiple Newton iteration steps have to be performed within each time step.

During each of these iteration steps, the system equations (1) and (2) need to be assembled. This means, based on the latest estimate of the degrees of freedom  $q$  and  $\dot{q}$  at the next time step, all system matrices etc. need to be calculated. Then the residues to these equations need to be evaluated. If the residues are not

within a user-defined precision, an increment needs to be calculated to improve the estimate of the degrees of freedom at the next time step. Both the assembly and the increment calculation are time-consuming. Even though the typical number of degrees of freedom is limited for common multibody models<sup>3</sup>, the computational burden of both system equations assembly and increment calculation is too heavy to allow for real-time calculations.

Real-time applications require converged results for a subset of the generalized coordinates, which will be called *output coordinates subset*  $q_{out}$ , at certain moments in time of the simulated process. Indeed, most real-time or faster-than-real-time applications don't require estimates of all values of  $q$  to correctly interact with other subsystems of the envisaged application. It is however not possible to simply ignore the other generalized coordinates  $q$  not belonging to  $q_{out}$ , as all generalized coordinates are needed to accurately model the system dynamics. It would thus be desirable to describe the system with less degrees of freedom, while still being able to describe the dominant dynamic effects of the coordinates  $q_{out}$ .

Furthermore, the used simulation technique should guarantee to produce converged simulation results  $q_{out}(t_{i+1})$  within a time shorter than the simulated time step  $h_i = t_{i+1} - t_i$ . The *simulation time*, the time needed to perform the calculation of these converged results, has to be shorter than the *simulated time*, the time span of the simulated process. Depending on the application requirements, this condition has to hold for each time step or, less stringent, the inequality should hold on average. The solving of a set of differential-algebraic equations (DAE), irrespective whether an explicit or an implicit solver is used, requires an unknown number of iterations per time step. If too much iterations are required to obtain convergence, the iteration will be forced to stop before having converged to keep up with the real-time process. The solving of a set of ordinary differential equations by an explicit solver on the other hand, requires a single step per time step, leading to a known number of computational operations to be performed per time step. Therefore, it is desirable to transform the set of differential algebraic equations into a set of ordinary differential equations.

For any application it would be useful if the original set of DAE's with  $n$  differential equations,  $m$  algebraic constraints and thus  $n - m$  intrinsic degrees of freedom, could be recast as a set of  $n - m$  ODE's and/or if the different equations would be less coupled such that increment calculation becomes less expensive. An approximation of the system which describes the system with far less degrees of freedom would also be desirable, preferably as an ODE. However, a DAE of a lesser dimension could also be solved faster than the original system. In this work the latter solution will be investigated. In future work, the author's focus will be shifted towards recasting the DAE-model into an ODE-model.

Current real-time running multibody models are typically strongly simplified. It is important to note that those simplifications mostly boil down to neglecting certain phenomena of the system. The quality of this approach thus heavily depends on engineering insight. Other techniques try to reduce the number of degrees of freedom by choosing the optimal choice of coordinates: using relative coordinates usually lead to fewer degrees of freedom [7]. In summary, one can conclude that none of the current model reduction techniques make use of the mathematical structure of the initial model. The goal of this paper is to offer a theoretical framework to develop a model reduction technique based on the mathematical structure of the model.

### 3 Projection upon a new coordinate system

The equations of motion expressed in the original coordinates  $q$  can be projected upon a new coordinate system  $\eta$ , through the transformation:

$$q = q(\hat{\eta}) \quad (7)$$

The coordinates  $\hat{\eta}$  are mapped from a space  $\Omega^{\hat{\eta}}$  upon  $\mathcal{R}^n$ .

---

<sup>3</sup>Typically 100's-1000's of DOFs

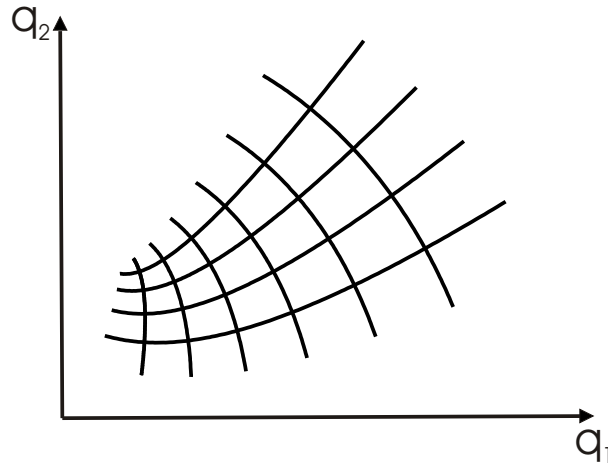


Figure 1: An example of curvilinear coordinates. The curved lines are lines for which only one coordinate  $(\eta)_i$  varies.

If only the time evolution of  $q_{out}$  and  $\dot{q}_{out}$  need to be reconstructable after the coordinate transformation, the following conditions apply to the mapping for it to be a regular coordinate transformation:

- It defines a bijection between  $\Omega_{\hat{\eta}}$  and  $\widehat{\Omega}_q^{ft}$
- It is continuously differentiable in  $\Omega_{\hat{\eta}}$
- The Jacobian of the transformation has maximal rank

If these conditions are met, the equations of motion expressed in the new coordinate system will take the following form:

$$M^{\hat{\eta}\hat{\eta}}(\hat{\eta}) \ddot{\hat{\eta}} + h^{\hat{\eta}}(\hat{\eta}, \dot{\hat{\eta}}) + \mathcal{V}_{,\hat{\eta}} + \Phi_{,\hat{\eta}}^T \lambda = g^{\hat{\eta}} \tag{8}$$

$$\Phi(q(\hat{\eta})) = 0 \tag{9}$$

with

$$\mathcal{V}_{,\hat{\eta}} = q_{,\hat{\eta}}^T \mathcal{V}_{,q} \tag{10}$$

$$g^{\hat{\eta}} = q_{,\hat{\eta}}^T g^q \tag{11}$$

$$M^{\hat{\eta}\hat{\eta}} = q_{,\hat{\eta}}^T M^{qq} q_{,\hat{\eta}} \tag{12}$$

$$(h^{\hat{\eta}})_i = (\Gamma^{\hat{\eta}\hat{\eta}\hat{\eta}})_{ijk} \dot{\hat{\eta}}_j \dot{\hat{\eta}}_k \tag{13}$$

The components of the new Christoffel symbol are easily obtained from (4) and (12):

$$(\Gamma^{\hat{\eta}\hat{\eta}\hat{\eta}})_{ijk} = \frac{\partial q_u}{\partial \hat{\eta}_i} \frac{\partial q_v}{\partial \hat{\eta}_j} \frac{\partial q_w}{\partial \hat{\eta}_k} (\Gamma^{qqq})_{uvw} + \frac{\partial q_u}{\partial \hat{\eta}_i} \frac{\partial^2 q_v}{\partial \hat{\eta}_j \partial \hat{\eta}_k} (M^{qq})_{uv} \tag{14}$$

### 3.1 Tangent to the constraint equations?

If both the values of  $q$  and  $\dot{q}$  as well as the reaction forces  $\Phi_{,q}^T \lambda$  should be reconstructable after performing the transformation, the following conditions apply to the mapping for it to be a regular coordinate transformation:

- It defines a bijection between  $\Omega_{\hat{\eta}}$  and  $\mathbb{R}^n$
- It is continuously differentiable in  $\Omega_{\hat{\eta}}$
- The Jacobian of the transformation has maximal rank, i.e.  $n$

Lagrange multipliers fulfill a role which is more than a mere mathematical way to impose the constraint equations (2). The term  $\Phi_{,q}^T \lambda$  in equation (1) represents the reaction forces in the connections of the mechanism. Often these reaction forces are of interest and thus need to be calculated. If all information about the reaction forces needs to be reconstructed after the transformation (7)  $\Phi_{,\hat{\eta},q}^T \hat{\eta}$  in equation (8) needs to span the same space as the matrix  $\Phi_{,q}^T$  in equation (1)<sup>4</sup>. Furthermore, the matrix  $\Phi_{,\hat{\eta}}^T$  needs to be sufficiently well conditioned to enable an accurate calculation of all reaction force terms  $\Phi_{,q}^T \lambda$  in equation (1)

If reaction forces are not of interest, it is wise to choose the new coordinate system in such a way that the matrix  $\Phi_{,\hat{\eta}}^T$  is as much rank deficient as possible for all configurations, i.e. equal to a null matrix. This implies that the coordinate axes locally lie in the tangent space to the flexible configuration space  $\hat{\Omega}_q^{ft}$  at each configuration. If this is the case, the change in violation of the constraint equation will thus be zero, for an arbitrary change of  $\hat{\eta}$ :

$$\Phi(q(\hat{\eta}_2)) - \Phi(q(\hat{\eta}_1)) = \int_{q(\hat{\eta}_1)}^{q(\hat{\eta}_2)} \Phi_{,q} dq = \int_{\hat{\eta}_1}^{\hat{\eta}_2} \Phi_{,q} q(\hat{\eta})_{,\hat{\eta}} d\hat{\eta} = 0 \quad (15)$$

Furthermore, if the first point  $q(\hat{\eta}_1)$  satisfies the constraint equations (2), the projected constraint equations (9) will thus automatically be satisfied. The initial set of differential algebraic equations (DAE), (1) and (2), has become a set of ordinary differential equations (ODE). The system will now be described by a minimal set of coordinates  $\hat{\eta}$ , i.e.  $n - m$  coordinates. ODE's are easier to solve and more suited to real-time solution as explained in section 2.3.

Brüls et al. [1], [2] proposes a coordinate transformation technique for which the new coordinate axes system lies within the tangent space to the flexible configuration space at each configuration. The coordinate transformation is however only valid away from certain singularities of the system: the actuator dead points. The coordinate transformation can lose its bijection property because of the used linearization as will be explained later in this paper, although the resulting error will be acceptable for small amplitude deformations.

If the coordinates  $\hat{\eta}$  are defined within the tangent space of  $\hat{\Omega}_q^{ft}$ , and if the reaction force due to a specific constraint equation is of interest, an additional coordinate can be added to the coordinate set  $\hat{\eta}$ . The most wise choice is to lay this additional coordinate along the gradient of this constraint equation. The projected set of equations will inevitably become a DAE again, but the projected set of equations will contain the least projected constraint equations as possible to be able to reconstruct the desired reaction force. In case of  $m_{desired}$  multiple desired reaction forces,  $m_{desired}$  additional coordinate axes should be chosen which locally and at each configuration span the same space as the gradients of the constraint equations of which the reaction forces need to be calculated. In this work reaction forces will however be assumed not to be of interest.

<sup>4</sup>For well-constructed models the constraint equations are linearly independent for all configurations. The rank of  $\Phi_{,q}^T$  is full, i.e. equal to the number of constraint equations.

### 3.2 Dependent and independent dynamics configuration space

Although the previous paragraph shows obvious advantages for the new coordinate system to be tangent to the constraint equations, it will further be assumed that this is not necessarily true for reasons of generality.

The new coordinate system is now chosen in such a way that  $\hat{\eta}$  can be decomposed as follows:

$$\hat{\eta} = \begin{bmatrix} \theta \\ \hat{\delta} \end{bmatrix} \tag{16}$$

The equations of motion (8) and (9) can now be decomposed into equations in  $\theta$  and equations in  $\hat{\delta}$  which are mutually coupled:

$$\begin{bmatrix} M^{\theta\theta}(\theta, \hat{\delta}) & M^{\theta\hat{\delta}}(\theta, \hat{\delta}) \\ M^{\hat{\delta}\theta}(\theta, \hat{\delta}) & M^{\hat{\delta}\hat{\delta}}(\theta, \hat{\delta}) \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\hat{\delta}} \end{bmatrix} + \begin{bmatrix} h^\theta(\theta, \hat{\delta}, \dot{\theta}, \dot{\hat{\delta}}) \\ h^{\hat{\delta}}(\theta, \hat{\delta}, \dot{\theta}, \dot{\hat{\delta}}) \end{bmatrix} + \begin{bmatrix} \mathcal{V}_{,\theta}(\theta, \hat{\delta}) \\ \mathcal{V}_{,\hat{\delta}}(\theta, \hat{\delta}) \end{bmatrix} + \begin{bmatrix} \Phi_{,\theta}^T \\ \Phi_{,\hat{\delta}}^T \end{bmatrix} \lambda = \begin{bmatrix} g^\theta \\ g^{\hat{\delta}} \end{bmatrix} \tag{17}$$

$$\Phi(q(\begin{bmatrix} \theta \\ \hat{\delta} \end{bmatrix})) = 0 \tag{18}$$

Similar to the equations (10), (11), (12), (13) and (14), the following equations explain the different terms in the equations (17) and (18):

$$\mathcal{V}_{,\theta} = q_{,\theta}^T \mathcal{V}_{,q} \tag{19}$$

$$\mathcal{V}_{,\hat{\delta}} = q_{,\hat{\delta}}^T \mathcal{V}_{,q} \tag{20}$$

$$g^\theta = q_{,\theta}^T g^q \tag{21}$$

$$g^{\hat{\delta}} = q_{,\hat{\delta}}^T g^q \tag{22}$$

$$M^{\theta\theta} = q_{,\theta}^T M^{qq} q_{,\theta} \tag{23}$$

$$M^{\theta\hat{\delta}} = q_{,\theta}^T M^{qq} q_{,\hat{\delta}} \tag{24}$$

$$M^{\hat{\delta}\theta} = q_{,\hat{\delta}}^T M^{qq} q_{,\theta} = (M^{\theta\hat{\delta}})^T \tag{25}$$

$$M^{\hat{\delta}\hat{\delta}} = q_{,\hat{\delta}}^T M^{qq} q_{,\hat{\delta}} \tag{26}$$

$$\begin{bmatrix} h^\theta(\theta, \hat{\delta}, \dot{\theta}, \dot{\hat{\delta}}) \\ h^{\hat{\delta}}(\theta, \hat{\delta}, \dot{\theta}, \dot{\hat{\delta}}) \end{bmatrix}_i = (h^{\hat{\eta}})_i = (\Gamma^{\hat{\eta}\hat{\eta}})_{ijk} \dot{\eta}_j \dot{\eta}_k \tag{27}$$

Now the motion of the system  $q$  is decomposed in a contribution within a subspace referred to as the *independent dynamics configuration space*  $\Omega_q^{\theta t} \subseteq \widehat{\Omega}_q^{ft}$  and a deviation from the independent dynamics configuration space into the *dependent dynamics configuration space*  $\Omega_q^{\hat{\delta} t} = \widehat{\Omega}_q^{ft} \setminus \Omega_q^{\theta t}$ .

Again, the goal of all this is to make the model cheaper to solve. This will be achieved by adopting the following assumption:

**The dynamics governing the contribution of the motion in the independent dynamics configuration space can be accurately approximated to be independent of the contribution within the dependent dynamics configuration space.**

The dynamics governing the remainder of the motion, all deviation from the independent dynamics configuration space into the dependent dynamics configuration space, will not be approximated as independent from the contribution within the independent dynamics configuration space. Actually, a further hierarchy is even possible if one assumes a second part of the dynamics to depend only on the first part, but to be independent of the third part and so on. This further hierarchy will not be considered in this paper.

This assumption results in a set of less coupled equations, which makes them easier to solve. An approximation error is made since some couplings are ignored. The quality of this approximation thus heavily depends upon the selection of the dependent and independent dynamics configuration spaces, or in other phrasing, on the couplings which will be ignored. This selection requires considerable engineering insight.

The independent dynamics configuration space could be defined by an equation:

$$\Theta(q) = \text{constant} \quad (28)$$

In more mathematical terms, the assumption becomes: the dynamics of all motion along isosubspaces<sup>5</sup> of the function  $\Theta(q)$  is approximated as being independent of the motion orthogonal to the isolines of the function  $\Theta(q)$ .

The contribution within the independent dynamics configuration space is modelled by the subset of coordinates  $\theta$ . The deviation from the independent dynamics configuration space into the dependent dynamics configuration space is modeled by the coordinates  $\hat{\delta}$ . This implies that the coordinate axes of the coordinates  $\theta$  locally span the same subspace as the tangents to the isosubspace of the function  $\Theta(q)$  at each configuration:

$$\Theta(q(\begin{bmatrix} \theta \\ \hat{\delta} \end{bmatrix})),_{\theta} = 0 \quad (29)$$

The coordinates  $\hat{\delta}$  on the other hand, are chosen in such a way that the jacobian  $\Theta(q(\begin{bmatrix} \theta \\ \hat{\delta} \end{bmatrix})),_{\hat{\delta}}$  has full rank at each configuration. This means, none of the the coordinate axes  $\hat{\delta}$  are tangent to the independent dynamics configurations space. As the movement along isosubspaces of  $\Theta(q)$  is fully characterized by a change of the coordinates  $\theta$ , with the coordinates  $\hat{\delta}$  being constant, the coordinates  $\hat{\delta}$  have a constant value along isosubspaces of  $\Theta(q)$ . As this constant can be shifted arbitrarily, the coordinates  $\hat{\delta}$  are chosen to be zero in the independent dynamics configuration space.

In order to make maximal use of the assumption to make time integration of equation (17) and (18) cheaper, the coordinates  $\theta$  and  $\hat{\delta}$  should be chosen in such a way that the equations in  $\theta$  and  $\hat{\delta}$  can be maximally decoupled. This can be achieved in the following way:

- The mapping  $q^{\theta} : \Omega_{\theta} \rightarrow \Omega_q^{\theta t}, \theta \mapsto q = q(\begin{bmatrix} \theta \\ 0 \end{bmatrix})$  is a regular coordinate transformation:
  - It is a bijection from  $\Omega_{\theta}$  to  $\Omega_q^{\theta t}$
  - It is continuously differentiable in  $\Omega_{\theta}$
  - The jacobian of the transformation has maximal rank, i.e. the dimension of the space  $\Omega_q^{\theta t}$
- In order to make maximal use of the assumed independence, the two coordinate subsets should be defined in such a way that their equations are maximally decoupled:

<sup>5</sup>The more-dimensional equivalent of an isoline.

- The coordinate axes of the coordinates  $\hat{\delta}$  must be chosen mass-orthogonal to the coordinate axes of the coordinates  $\theta$  in each configuration:

$$M^{\hat{\delta}\theta} = (M^{\theta\hat{\delta}})^T = 0 \quad (30)$$

- The coordinates  $\theta$  and  $\hat{\delta}$  are also coupled through the constraint equations. In order to get rid of this coupling the coordinate axes of the coordinates  $\hat{\delta}$  should be chosen tangent to the constraint equations.

$$\Phi_{,qq} \left( \begin{bmatrix} \theta \\ \hat{\delta} \end{bmatrix} \right)_{,\hat{\delta}} = 0 \quad (31)$$

Due to this property the projected constraint equations (18) degenerate:

$$\Phi \left( q \left( \begin{bmatrix} \theta \\ \hat{\delta} \end{bmatrix} \right) \right) - \Phi \left( q \left( \begin{bmatrix} \theta \\ 0 \end{bmatrix} \right) \right) = \int_{q(\theta,0)}^{q(\theta,\hat{\delta})} \Phi_{,qq} dq = \int_0^{\hat{\delta}} \Phi_{,qq} \left( \begin{bmatrix} \theta \\ \tilde{\delta} \end{bmatrix} \right)_{,\tilde{\delta}} d\tilde{\delta} = 0 \quad (32)$$

The integral in equation (32) is path independent, as the vector field  $\Phi_{,q}$  is defined as the gradient of a singularity-free function.

- The total transformation still has to be a regular transformation:

- It is a bijection  $(\theta, \hat{\delta}) \mapsto q = q \left( \begin{bmatrix} \theta \\ \hat{\delta} \end{bmatrix} \right)$
- It is continuously differentiable in  $\Omega_\theta \times \Omega_{\hat{\delta}}$
- Its jacobian has full rank

The assumption that the dynamics governing the contribution of the independent dynamics configuration space, modeled by the coordinates  $\theta$ , is independent of the remaining part of the motion of the system, modeled by the coordinates  $\hat{\delta}$ , has the following consequences on terms in equations (17):

- $M^{\theta\theta}(\theta, \hat{\delta})$  and  $\mathcal{V}_{,\theta}$  only depend on  $\theta$ , and not on  $\hat{\delta}$ .
- $h^\theta$  only depends on  $\theta$  and  $\dot{\theta}$ , not on  $\hat{\delta}$  nor  $\dot{\hat{\delta}}$ .

The equations of motion (17) and (18) can now be rewritten in the following decoupled way:

$$M^{\theta\theta}(\theta)\ddot{\theta} + h^\theta(\theta, \dot{\theta}) + \mathcal{V}_{,\theta}(\theta) + \Phi_{,\theta}^T \lambda = g^\theta \quad (33)$$

$$\Phi \left( q \left( \begin{bmatrix} \theta \\ 0 \end{bmatrix} \right) \right) = 0 \quad (34)$$

$$M^{\hat{\delta}\hat{\delta}}(\theta, \hat{\delta})\ddot{\hat{\delta}} + h^{\hat{\delta}}(\theta, \hat{\delta}, \dot{\hat{\delta}}) + \mathcal{V}_{,\hat{\delta}}(\theta, \hat{\delta}) = g^{\hat{\delta}} \quad (35)$$

with

$$h^\theta(\theta, \dot{\theta}) = (\Gamma^{\theta\theta\theta})_{ijk} \dot{\theta}_j \dot{\theta}_k \quad (36)$$

The components of the Christoffel symbol  $\Gamma^{\theta\theta\theta}$  can be easily obtained by neglecting the terms in  $\hat{\delta}$  in (14).

$$(\Gamma^{\theta\theta\theta})_{ijk} = \frac{\partial q_u}{\partial \theta_i} \frac{\partial q_v}{\partial \theta_j} \frac{\partial q_w}{\partial \theta_k} (\Gamma^{qqq})_{uvw} + \frac{\partial q_u}{\partial \theta_i} \frac{\partial^2 q_v}{\partial \theta_j \partial \theta_k} (M^{qq})_{uv} \quad (37)$$

During time integration of the set of equations (33), (34) and (35), the equations (33) and (34) can be solved independently of (35). Equation (35) will then use the values of  $\theta$  as input to perform the time integration of this equation to obtain the values of  $\hat{\delta}$ . In a final step, the back transformation can be performed to get the values of  $q$  in the time domain. Because of the assumption there will be some loss of precision. This loss of precision will depend solely on the partition between the independent dynamics configuration space and the dependent configuration space.

## 4 Decomposition into large rigid body motion and small amplitude deformation

For the larger part of the load cases of most real-life mechanisms, the overall motion  $q$  can be decomposed into a large rigid body motion  $q^r \in \Omega_q^{rt}$ , the motion of the mechanism without deformation, and a small amplitude deformation  $q^f$ .

$$q = q^r + q^f \quad (38)$$

The dynamics of the rigid body motion can be approximated acceptably as being independent from the small amplitude deformation. The independent dynamics configuration space  $\Omega_q^\theta$  will thus be chosen equal to the rigid configuration space  $\Omega_q^{rt}$ . The dependent dynamics configuration space will be chosen equal to the deformed configuration space  $\Omega_q^{dt}$ . Thus  $\Theta(q) = \mathcal{V}$ .

As a rigid body motion doesn't change the potential energy due to deformation,  $\mathcal{V}_\theta = 0$  in (33).

The equations of motion in terms of  $\theta$  and  $\hat{\delta}$  become:

$$M^{\theta\theta}(\theta)\ddot{\theta} + h^\theta(\theta, \dot{\theta}) + \Phi_{,\theta}^T \lambda = g^\theta \quad (39)$$

$$\Phi(q\left(\begin{array}{c} \theta \\ 0 \end{array}\right)) = 0 \quad (40)$$

$$M^{\hat{\delta}\hat{\delta}}(\theta, \hat{\delta})\ddot{\hat{\delta}} + h^{\hat{\delta}}(\theta, \delta, \dot{\theta}, \hat{\delta}) + \mathcal{V}_{,\hat{\delta}}(\theta, \hat{\delta}) = g^{\hat{\delta}} \quad (41)$$

Up till now, only the properties of the coordinate transformation have been discussed. Nothing has been said about how the new coordinate system should be chosen, or whether it is actually possible to define a coordinate transformation having these properties for a given initial model. In practice the rigid body motion can be modeled using any formulation for the rigid body mechanism, e.g. relative, Cartesian or nodal coordinates, and the coordinates  $\theta$  can simply be chosen the same as the coordinates of the initial rigid body model. It is of course also possible to define a new set of coordinates. Brüls et al. [2] defines a minimal coordinates mapping based on an advanced form of coordinate elimination. The approach proposed by Brüls et al. however fails near the system's actuator singularities. In a later stage, a singularity-free modal parametrization will be investigated by the author.

It will further be assumed that, for a given value of  $\theta$ , the motion can be linearized around the undeformed state  $q^r$ . This assumption can be justified by the small amplitude of the elastic deformation. The result of this assumption is that  $M^{\hat{\delta}\hat{\delta}}(\theta, \hat{\delta})$  only depends on  $\theta$ .  $\mathcal{V}_{,\hat{\delta}}(\theta, \hat{\delta})$  can be written as  $K^{\hat{\delta}\hat{\delta}}(\theta)\hat{\delta}$ .  $\frac{\partial^2 q_v}{\partial \hat{\eta}_j \partial \hat{\eta}_k}$  of equation (14) thus gets zero if both  $\hat{\eta}_j$  and  $\hat{\eta}_k$  belong to the  $\hat{\delta}$  partition (7) of the coordinates  $\eta$ .

$$\frac{\partial^2 q_v}{\partial \hat{\delta}_j \partial \hat{\delta}_k} = 0 \quad (42)$$

It is important to note that the transformation might lose its bijection-property: several values  $(\theta, \hat{\delta})$  might map to the same value of  $q$ , but each value of  $(\theta, \hat{\delta})$  will be mapped to a unique value  $q = q^r + q^f$ . This is shown in figure 2. Because of the linearization around  $q^r$ ,  $q^f$  will be approximated. The error will be acceptable due to the small deformation. As long as the system is described in terms of the coordinates  $(\theta, \hat{\delta})$  no problem will occur, because the back transformation to  $q$  is always unique, albeit with a small error.

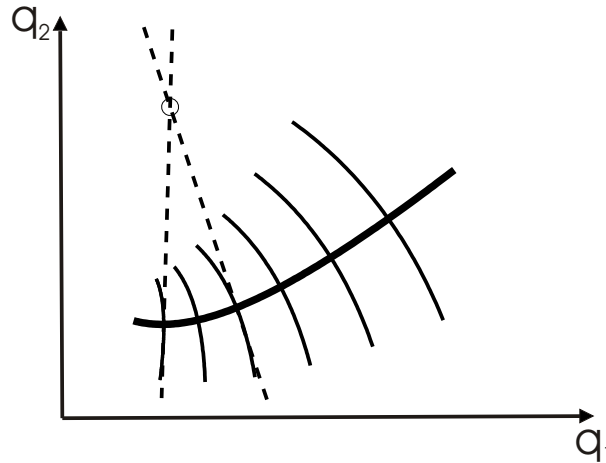


Figure 2: The bijection property can be lost because of the linearization around undeformed states. The thick line represents the rigid configuration space. The coordinate axes of the coordinates  $\hat{\delta}$  are mass-orthogonal to this space (for the sake of visibility  $M = I$  is assumed). The dashed lines represent the linearized coordinate axes of the coordinates  $\hat{\delta}$ . Due to the linearization multiple values of  $(\theta, \hat{\delta})$  map to the point marked by the circle. Each value of  $(\theta, \hat{\delta})$  however maps to a unique value of  $q$ .

## 5 Practical use of the coordinate transformation

As stated in the introduction, the goal of all these manipulations is to end up with systems of equations which have the capability to faster produce converged results for  $q$  and its time derivatives. Therefore, the overall process that needs to be performed to obtain simulation results for a given period should be considered. The overall process that needs to be performed each time step consists of possibly several iteration steps, each consisting of:

1. Assembly of the system equations
2. Increment calculation

A gain in simulation speed can thus be achieved by:

- Limiting the number of iteration steps per time step. This can be achieved by converting the DAE into an ODE by choosing all of the new coordinate axes to be tangent to the constraint equations and using an explicit solver.
- Performing the assembly of the systems equations before the actual simulation phase.
- Rendering the increment calculation cheaper. This can be achieved by ignoring the non-relevant part of the dynamics and thus reducing the number of degrees of freedom and dimension of the set of equations.

A gain in simulation speed could also be achieved by improving the DAE/ODE-solver and optimizing the time step. This is not considered in this work.

It is important to note that unless for very simple models, it will be practically impossible to define an analytical coordinate transformation which is able to reach this goals. Therefore, we seek for a coordinate transformation for which the properties (bijection, differentiable, full rank of Jacobian) can be proven and for which the mapping in both directions ( $\hat{\eta} \mapsto q$  and  $q \mapsto \hat{\eta}$ ), and all elements of the projected model can be calculated up to an arbitrary precision. The mapping and the elements of the projected model can thus only be evaluated numerically. The mapping and the elements of the projected model will be calculated for a discrete set of points in the independent dynamics subspace. During simulation, the mapping and the elements of the projected model will be interpolated between nearby previously calculated points. As this calculation can be performed before the actual simulation, it is allowed to be computationally expensive. All these results will have to be stored, thus limiting memory requirements is also of importance.

The method described below considers the set of  $\theta$ -coordinates to describe the rigid body motion. Their corresponding coordinate axes do not necessarily need to be tangent to the constraint equations. It can be a redundant set of degrees of freedom. The method can thus be split up in two processes: a preparation phase, which is allowed to be computationally expensive, and a simulation phase, during which simulation speed is of utmost importance.

## 5.1 Preparation phase

The preparation comprises several substeps:

1. Define a set of points within the independent dynamics configuration space in which the coordinate axes for the  $\hat{\delta}$ -coordinates, and the elements of the reduced model will be calculated. Later, during the simulation phase, this data will be interpolated to obtain reduced models for points in between the chosen set of points. The selection of this set of points is thus a compromise between computational burden during the preparation phase and the interpolation error during the simulation phase.
2. Choose the number of coordinates in the  $\delta$ -set and their definition. They should be defined in such a way that the following properties hold: bijection, continuously differentiable, full rank Jacobian.
3. For each point within the discretized set of points within the independent dynamics configuration space:
  - (a) Define the value of  $\theta$  (this is trivial if no coordinate transformation is performed on the rigid body coordinates)
  - (b) Calculate the value of  $\delta$ : for the sake of simplicity, it will be assumed that  $\hat{\delta} = 0$  on the independent dynamics configuration space
  - (c) Calculate and store the elements of the reduced model
  - (d) (optionally) Calculate the gradient of the reduced model elements to check whether the discretization of the independent dynamics manifold should be finer or is allowed to be coarser

At the end of this process a database exists in which the elements of the reduced model can be interpolated based on the value of  $\theta$ .

## 5.2 Simulation phase

At each time step the values of the coordinates  $(\theta, \hat{\delta})$  need to be defined at the next time  $t_{i+1}$  point based on the known values of  $(\theta, \hat{\delta})$  at the current time step  $t_i$ . To do so, each time step the following process needs to be performed:

- Solve for the new values of  $\theta(t_{i+1})$  and  $\dot{\theta}(t_{i+1})$  (equations (39) and (40)).
- Interpolate between reduced models of nearby previously calculated points based on  $\theta(t_i)$  and/or  $\theta(t_{i+1})$
- Integration over time step to obtain  $\hat{\delta}(t_{i+1})$  and  $\dot{\hat{\delta}}(t_{i+1})$  (equation (41))
- (optionally) Retransform to the subset of physical coordinates of interest  $q_{out}(t_{i+1})$  and  $\dot{q}_{out}(t_{i+1})$

## 6 Meeting the goal: faster simulation

The set of equations (8) and (9) might be easier to solve than the original equations (1) and (2), because it is possible to describe the system with less degrees of freedom if the new coordinate axes are chosen tangent to the flexible configuration space. Furthermore, the system matrices etc. could be cheaper to manipulate because they might be more sparse and/or less coupled if the degrees of freedom are chosen wisely. The set of equations (39), (40) and (41) might be still cheaper to solve because of the decoupling assumption.

These sets of equations all use at least  $n - m$  degrees of freedom. Up till now, all dynamic properties of the initial model are still present in the model described in the new coordinate system, except for some ignored couplings. A significant further improvement can only be achieved by decreasing the number of degrees of freedom, thus by neglecting some of the dynamics of the system and unavoidably sacrifice some accuracy. The goal is to neglect and/or approximate the least relevant part of the dynamics and to find the optimal trade-off between gain in computational speed and loss of accuracy.

Although possible, it is not advisable to approximate the rigid body dynamics (and by extension the independent dynamics), due to the high non-linear dynamics of the rigid body dynamics. Furthermore, the elastic deformation dynamics depend on these dynamics in a highly non-linear manner. Approximation of the independent dynamics will not be considered in this work.

Significant gain in simulation speed can be achieved by approximating the dynamics of the elastic deformation. Different techniques exist to approximate dynamical systems. The approximation techniques considered here are designed to take advantage of the mathematical structure of the model. Common ways for mathematics-based model reduction techniques are: Singular Value Decomposition with truncation of the eigenmodes and approximation of the omitted eigenvectors, Krylov vectors and neural networks [3].

Unlike SVD-based techniques Krylov vectors and neural networks approximation techniques depend on the experiments for which the resulting model is tailored to optimally approximate the initial model. Actually they don't use the mathematical structure of the model advantageously. Singular Value Decomposition techniques are chosen because they of the following properties:

- Cheap equations assembly, solving and back transformation
- Minimal loss of precision which can be quantified
- Limited memory requirements to store the reduced models calculated in the preparation phase
- The reduced equation take the same form as the initial set of equations. They can thus be solved by the same solvers. This property is useful for further developments.

### 6.1 Definition of the coordinates $\hat{\delta}$

In linear structural dynamics it is common practice to approximate the dynamic properties of a system within a certain frequency band by the contribution of the eigenmodes with eigenfrequencies near or within this

frequency band and specific other deformation patterns to approximate the contribution to the system's dynamics of the omitted eigenfrequencies.

The coordinate subsystem  $\hat{\delta}$  can be decomposed as follows:

$$\hat{\delta} = \begin{bmatrix} \delta \\ \delta^\epsilon \end{bmatrix} \quad (43)$$

The equation (41) thus gets the following form:

$$\begin{bmatrix} M^{\delta\delta}(\theta, \hat{\delta}) & M^{\delta\delta^\epsilon}(\theta, \hat{\delta}) \\ M^{\delta^\epsilon\delta}(\theta, \hat{\delta}) & M^{\delta^\epsilon\delta^\epsilon}(\theta, \hat{\delta}) \end{bmatrix} \begin{bmatrix} \ddot{\delta} \\ \ddot{\delta}^\epsilon \end{bmatrix} + \begin{bmatrix} h^\delta(\theta, \hat{\delta}, \dot{\theta}, \dot{\hat{\delta}}) \\ h^{\delta^\epsilon}(\theta, \hat{\delta}, \dot{\theta}, \dot{\hat{\delta}}) \end{bmatrix} + \begin{bmatrix} K^{\delta\delta}(\theta) & K^{\delta\delta^\epsilon}(\theta) \\ K^{\delta^\epsilon\delta}(\theta) & K^{\delta^\epsilon\delta^\epsilon}(\theta) \end{bmatrix} \begin{bmatrix} \delta \\ \delta^\epsilon \end{bmatrix} + \begin{bmatrix} \Phi_\delta^T \\ \Phi_{,\delta^\epsilon}^T \end{bmatrix} \lambda = \begin{bmatrix} g^\delta \\ g^{\delta^\epsilon} \end{bmatrix} \quad (44)$$

In these equations the dependent dynamics are linearized around the corresponding undeformed configuration. Similar to Component Mode Synthesis originating from linear structural dynamics, the dynamics of the elastic deformation of the system will now be approximated by its dominant dynamic phenomena. These dominant dynamic phenomena will be modeled by the degrees of freedom  $\delta$ , the remaining part of the dynamics can be omitted.

The system described by  $\theta$  and  $\delta$  will not be able to describe all the possible configurations of the initial model. The set of coordinates  $(\theta, \delta)$  maps to a subspace of the flexible configuration space, which will be referred to as the dominant dynamics configurations space. The coordinate transformation is no longer a bijection upon the flexible configuration space as not all configurations can be described. They can however be described with an acceptable precision.

## 6.2 Modal representation of the flexibility

The flexibility of the system can be represented by modal vectors. It is important to note that the rigid body motion is not described by modal vectors, but by the possibly redundant set of coordinates  $\theta$ . Contrary to the often used modal representation of the flexibility of components in multibody models, the mode sets will now represent system-level deformation patterns. Component-level modal representation suffers from the fact that all those degrees of freedom on which a constraint equation (2) is imposed, need to be retained. By choosing system-level deformation patterns that respect the constraint equations, i.e. that are tangent to the flexible configuration space, those degrees of freedom do not need to be retained: because of the intrinsic respecting of the constraint equations, these degrees of freedom are not required for the solving of the projected system equations.

For the optimal modal representation, or in other phrasing, the optimal definition of the  $\delta$ -coordinate axes of the new coordinate system, the following properties are desired:

- The storing of the elements of the projected equations should require the least possible memory.
- The projected equations should be as cheap as possible to solve.
- The calculation of the reduced model should be as cheap as possible. This is less of an issue however, as the time constraints during the preparation phase are less stringent.

The first two goals can be achieved by choosing the mode set in such a way that the reduced mass and stiffness matrices are maximally diagonal. Storing a sparse matrix such as a diagonal matrix is much cheaper than storing a densely populated matrix. Furthermore, if some elements of the reduced mass and stiffness matrices

are the same regardless of the configuration, they only need to be stored once, and not for each configuration. As the mode set can be mass orthogonalized, the mode set can be defined in such a way that the reduced mass matrix becomes the identity matrix. If this is done consistently for all configurations, it only needs to be stored once. Diagonal reduced mass and stiffness matrices also lead to less coupled equations, which makes the solving cheaper. Although the mode set can be chosen in such a way that the reduced mass and stiffness take a simple form, it is impossible to reduce the complexity of the second term in equation (44) and the reduced mass and stiffness matrices simultaneously. Mathematically, it is however easier to calculate mass orthonormalized modal vectors, a classic eigenvalue problem, than to calculate modal vectors which simplify the second term in equation (44), which can be done by demanding that specific elements of the Christoffel symbol in equation (14) are equal to a certain value.

Craig defines multiple sets of dynamic component mode supersets [4]. These are sets of modes able to approximate the dynamics of a linear elastic structure up to a given frequency with the least number of modes and the least loss of precision. Amongst these different sets, the dynamic residual attachment-mode superset leads to the most sparse reduced mass and stiffness matrix, both of them being diagonal. Furthermore, by normalizing the modes with respect to the mass matrix, the reduced mass matrix becomes the identity matrix. Contrary to the dynamic residual attachment-mode superset Craig proposes, the rigid body motion will not be described by this mode set, as it is already described by the coordinates  $\theta$ . However, it is possible to use any superset to represent the flexibility of the mechanism. Bruls, whose focus wasn't on speeding up the simulation, proposes the Hurty set, which has less interesting properties for fast simulation.

The selected mode set should however, together with the coordinates  $\theta$  used to represent the rigid body motion, result in an allowed coordinate transformation. This leads to the following conditions for the mode set:

- The modal vectors should always be linearly independent of the rigid body modes. For computational efficiency, mass orthogonality to the rigid body modes is preferable.
- They should be continuously differentiable. The modal vectors are a function of the elements of the initial model equations (1) and (2), which vary smoothly within the flexible configuration space. Furthermore, if the functions defining the modal vectors are smooth within the respective domain, the modal vectors will be smooth as well. This is e.g. the case for the eigenmodes of the system.
- The modes should be linearly independent to obtain a coordinate transformation's Jacobian of full rank. This is the case for a well chosen mode set, e.g. the different mode supersets proposed by Craig [4].
- The change of the modal coordinates  $\delta$  should be path independent when going from one configuration to another. It still remains to be proven that this condition holds for an elastic deformation mode of an arbitrary flexible mechanism. This topic will be addressed by the author in the future.

Under the condition that the deformation remains small so that the approximation by linearization around the undeformed configuration is acceptable, the conditions mentioned above result in a regular coordinate transformation.

## 7 Conclusions and future work

Current simulation techniques are not powerful enough to allow for real-time simulation of flexible mechanisms. By describing the initial degrees of freedom in a wisely chosen curvilinear coordinate system, the resulting projected equations of motion can become much cheaper to solve. The overall process consists of defining the coordinate transformation, calculating the elements of the projected equation and solving the projected equations. This overall process is more expensive than solving the initial system equations. The

advantage of this methodology however is that an important part of the computational load can be shifted to a preparation phase, such that the computational load during the actual simulation phase is significantly reduced. By choosing the new coordinate axes to be locally tangent to the subspace allowed by the constraint equations, the initial set of differential-algebraic equations can be recast as a set of ordinary differential equations. By adopting some assumptions, the model can be significantly simplified. Firstly, for a lot of applications, certain degrees of freedom can be assumed to be independent of other degrees of freedom. By ignoring these weak couplings, the resulting equations are less coupled and thus cheaper to solve. Secondly, the dynamics of the system can be approximated by only considering the dominant dynamic properties and approximating the remainder of the dynamic properties.

Although this paper discusses a lot of desirable properties for a coordinate transformation, it still remains to be proven that it is possible to create a valid coordinate transformation that incorporates all these nice features. Currently, no singularity-free technique exist. In the near future the author will focus on defining a global singularity-free coordinate transformation incorporating those desirable properties, as well as applying this methodology to a test case.

## Acknowledgements

The research of Gert Heirman is funded by a Ph.D. grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

## References

- [1] O. Brüls, P. Duysinx, J-C. Golinval *The global modal parametrization for non-linear model-order reduction in flexible multibody dynamics*, International Journal for Numerical Methods in Engineering, vol. 69, no. 5, pp. 947-77, Jan 2007
- [2] O. Brüls, P. Duysinx, J-C. Golinval *A Model Reduction Method for the Control of Rigid Mechanisms*, Multibody System Dynamics, vol. 15, no. 3, pp. 213-227, Apr 2006
- [3] C. L. Bottasso, L. Riviello *Trim of rotorcraft multibody models using a neural-augmented model-predictive auto-pilot* Multibody System Dynamics, vol. 18, pp. 299-321, 2007
- [4] R.R. Craig *A review of time-domain and frequency-domain component mode synthesis methods* Journal of Modal Analysis, vol. 2, no. 2, pp. 59-72, Apr 1987
- [5] M. Géradin, A. Cardona *Flexible Multibody Dynamics: a finite element approach*, Wiley, New York, 2001
- [6] O. Wallrapp, R. Schwertassek *Representation of geometric stiffening in multibody system simulation*, International Journal for Numerical Methods in Engineering, 32(8):1833-1850, 1991
- [7] J. G. de Jalon, E. Bayo *Kinematic and dynamic simulation of multibody simulation the real-time challenge*, Springer, 1993