



**HAL**  
open science

## **A modular dynamic model for multibody system adapted to interactive simulation**

Nicolas Seguy, Pierre Joli, Zhi-Qiang Feng, Madeleine Pascal

► **To cite this version:**

Nicolas Seguy, Pierre Joli, Zhi-Qiang Feng, Madeleine Pascal. A modular dynamic model for multibody system adapted to interactive simulation. ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Sep 2003, Chicago, United States. pp.71-79, 10.1115/DETC2003/VIB-48311 . hal-01179534

**HAL Id: hal-01179534**

**<https://hal.science/hal-01179534>**

Submitted on 29 Jun 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A MODULAR DYNAMIC MODEL FOR MULTIBODY SYSTEM ADAPTED TO INTERACTIVE SIMULATION

**N. SEGUY**

Laboratoire Systèmes Complexes, Université d'Evry  
40 rue du Pelvoux, 91020 Evry France  
nseguy@iup.univ-evry.fr

**P. JOLI**

Laboratoire Systèmes Complexes, Université d'Evry  
40 rue du Pelvoux, 91020 Evry France  
pjoli@iup.univ-evry.fr

**Z.Q. FENG**

Laboratoire Mécanique d'Evry  
40 rue du Pelvoux, 91020 Evry France  
feng@iup.univ-evry.fr

**M. PASCAL**

Laboratoire Systèmes Complexes, Université d'Evry  
40 rue du Pelvoux, 91020 Evry France  
mpascal@iup.univ-evry.fr

## ABSTRACT

This paper presents a modular model of rigid multibody system using the acceleration-based augmented Lagrangian formulation. An important effort on the formulation of the governing equations has been made in order to meet the requirements for interactive simulation in computer aided design. Each body has been considered as an independent numerical component with its own numerical parameters, own mechanical parameters and own numerical integration scheme. Non-holonomic and holonomic constraints have been implemented in this formulation. This present work can be considered as an extended formulation of Bayo et al. [1] to the problem of interactive design and particular attention is paid to define the criteria of numerical stability.

**Keywords:** modular dynamics, stabilisation method, augmented Lagrangian, numerical components

## 1 INTRODUCTION

Traditionally, a mechanical engineer in charge of the design of a mechanism with some expected behaviour starts with a draft version and then performs a kinematic, static, and dynamic analysis. According to the results, he refines the mechanism and restarts another cycle of design. Introducing computer simulation accelerates this process considerably and for this reason a lot of mechanical

system simulation softwares exists (ADAMS, CATIA, SOLIDWORKS...). All these software programs have graphical user interfaces to help the design of mechanisms at the geometric level, sometimes at the static level, but rarely at the dynamical level. It is not possible, for example, to modify the model during the simulation by eliminating, introducing or modifying a component in the mechanism. It is necessary to follow the traditional cycle of modelling in these software programs which involves a pre-processing phase (to define the geometry and the loading), a solution phase (to solve the equations of motion) and a post-processing phase (to analyse the results). The major reason for this shortcoming is that the dynamic model solved is not associated to an object-oriented design. A primary condition to realise such objects consists in defining a dynamic model as a package of autonomous numerical components. Another condition is to use an object-oriented programming approach [6,7].

The purpose of this paper is to define a modular dynamic model using the acceleration-based augmented Lagrangian formulation [1]. In the second section, we present classic formulations to model a multibody system in a centralised manner. The advantages, referenced by other authors [3], of the acceleration-based augmented Lagrangian method, are presented. In the third section two criteria of numerical stability are defined relative to holonomic

and non-holonomic algebraic constraints. In the fourth section, the choice of generalised coordinates for a rigid body is discussed. The natural coordinates provide some interesting features, already referenced [10,11], and in our opinion the main interest is the facility of connection with finite element structures. In the fifth section, we present our modular dynamic model in acceleration-based augmented Lagrangian formulation and the concept of numerical components associated to a rigid body. In the sixth section, we present two numerical results using modular dynamic models. The first one compares the simulation of a double pendulum with the results obtained by a centralised dynamic model. The second one is a simple pendulum subject to topology changes pointing out the properties of modular modelling.

## 2 CENTRALISED MODEL

### 2.1 Classical Lagrangian formulation

The governing equations of a multibody system derived from Lagrange's equations [12,13] are given by:

$$\begin{cases} \frac{d}{dt} \left( \frac{\partial K}{\partial \dot{q}} \right) + \Phi_q^{hT} \lambda + \Phi_q^{nhT} \mu = Q + \frac{\partial K}{\partial q} \\ \phi^h(q, t) = 0 \\ \phi^{nh}(q, \dot{q}, t) = 0 \end{cases} \quad (1)$$

- $q$ ,  $\dot{q}$ ,  $\ddot{q}$  are respectively the generalised vector of  $n$ -coordinates, velocities and accelerations.
- $Q$  is the vector including internal and external generalised forces dependent on time  $t$ ,  $q$ ,  $\dot{q}$ .
- $K$  is the kinetic energy of the system.  $K = \sum K_i$  where  $K_i$  is the kinetic energy of one body.  $K$  has the following quadratic symmetrical form:  $K = \frac{1}{2} \dot{q}^T \mathbf{M}(q) \dot{q}$  where  $\mathbf{M}(q)$  is the semi-definite positive mass matrix of the system. This matrix is invertible, except for singular configurations.
- $\Phi_q^h = \frac{\partial \phi^h}{\partial q}$  and  $\Phi_q^{nh} = \frac{\partial \phi^{nh}}{\partial \dot{q}}$  are respectively the  $(m \times n)$  jacobian matrix associated to the vector of holonomic constraints  $\phi^h$  and the  $(r \times n)$  jacobian matrix associated to the vector of non holonomic constraints  $\phi^{nh}$ . The equations  $\phi^{nh} = 0$  constrain the generalised velocities and have the general form  $\Phi_q^{nh}(q) \dot{q} + a(t) = 0$ .

The governing equations can be written as:

$$\begin{cases} \mathbf{M}(q) \ddot{q} - R(q, \lambda, \mu) = Q(q, \dot{q}, t) + Q_c \\ \phi^h(q, t) = 0 \\ \phi^{nh}(\dot{q}, q, t) = 0 \end{cases} \quad (2)$$

where:

- $Q_c(q, \dot{q}, t) = \frac{1}{2} \dot{q}^T \frac{\partial \mathbf{M}}{\partial q} \dot{q} - \dot{\mathbf{M}} \dot{q}$  represents the generalised centrifugal and Coriolis forces,
- $R(q, \lambda, \mu) = -(\Phi_q^{hT} \lambda + \Phi_q^{nhT} \mu)$  represents the necessary generalised reaction forces needed to satisfy the constraints equations  $\phi^h = 0$  and  $\phi^{nh} = 0$ .

The system of Eq. (2) represents a system of  $(n+m+r)$  differential algebraic equations (DAE) of index 3. The solution of such a system can not be obtained directly. Two methods are usually followed:

- One method consists in transforming the system of DAE into a system of ordinary differential equations (ODE). Thus, it will be possible to use easily a classic numerical scheme to solve the problem.
- Another method is to transform the DAE into an associated incremental formulation. The incremental formulation is well suited to a numerical implicit resolution and offers good properties of numerical stability [4].

These transformations lead to lower index differential systems but can however introduce numerical instability. In the past, several stabilisation methods have been proposed such as GGL-stabilisation method [9], Park's stabilisation method [5], waveform relaxation method or perturbation method, see [14] for a review of these methods. In the following, some standard techniques suitable for modular models are described.

### 2.2 Substitution method with Baumgarte stabilisation [2]

To transform the previous system of DAE into a system of ODE of second order (index 1), we need to differentiate twice the holonomic constraint equations and once the non-holonomic constraint equations as follows:

$$\mathbf{M}(q) \ddot{q} - R(q, \lambda, \mu) = Q + Q_c \quad (3-a)$$

$$\Phi_q^h \ddot{q} = -\dot{\Phi}_q^h \dot{q} - \dot{\Phi}_t \quad (3-b)$$

$$\Phi_q^{nh} \ddot{q} = -\dot{\Phi}_q^{nh} \dot{q} - \dot{a} \quad (3-c)$$

This system is linear with respect to the  $(n+m+r)$  unknowns  $\ddot{q}$ ,  $\lambda$  and  $\mu$ . From Eq. (3-a), it is possible to compute  $\ddot{q}$  in terms of  $\lambda$  and  $\mu$ , then after substitution into Eq. (3-b) and Eq. (3-c),  $\lambda$  and  $\mu$  are computed as a function of  $q$  and  $\dot{q}$ . It remains then to substitute  $\lambda$  and  $\mu$  by this last function into Eq. (3-a) to obtain the following ODE:

$$\mathbf{M}\ddot{q} = \mathcal{Q}_T - \Phi^T \mathbf{B}^{-1} (\Phi \mathbf{M}^{-1} \mathcal{Q}_T + f) \quad (4)$$

with

- $\Phi = \begin{bmatrix} \Phi_q^h \\ \Phi_q^{nh} \end{bmatrix} \quad \mathbf{B} = \Phi \mathbf{M}^{-1} \Phi^T$
- $f = \begin{Bmatrix} \dot{\Phi}_q^h \dot{q} + \dot{\Phi}_t \\ \dot{\Phi}_q^{nh} \dot{q} + \dot{a} \end{Bmatrix} \quad \mathcal{Q}_T = \mathcal{Q} + \mathcal{Q}_c$

If standard numerical integration procedures are used to solve the above ODE, the geometrical constraints are not satisfied after a short time of simulation (drifting constraints). The resulting constraint violations are commonly stabilised using Baumgarte's method which consists in modifying  $f$  in the following way:

$$\tilde{f}(\alpha, \beta, \gamma) = f + \begin{Bmatrix} \alpha \phi^h + \beta \dot{\phi}^h \\ \gamma \phi^{nh} \end{Bmatrix} \quad (5)$$

where  $\alpha$ ,  $\beta$ ,  $\gamma$  are stabilisation constants which avoid large violations on the constraints  $\phi^h = 0$ ,  $\dot{\phi}^h = 0$  and  $\phi^{nh} = 0$ .  $\alpha$ ,  $\beta$ ,  $\gamma$  are arbitrarily chosen but dependent however on the multibody system and on external forces. In this work, for a sake of simplicity, the same range of parameters  $\alpha$ ,  $\beta$ ,  $\gamma$  for all holonomic and non-holonomic constraints are used, although, in general cases, they can be different.

In the case of singular positions or redundant constraints, the constraint equations are dependent, the matrix  $\mathbf{B}$  is no longer invertible and a bifurcation or a chaotic behaviour of the solution may appear.

### 2.3 Penalty formulation

Another approach to transform a system of DAE into a system of ODE consists in defining an explicit formulation of the Lagrange multipliers using a penalty function:

$$\lambda = p_h \phi^h + v \dot{\phi}^h \quad (6-a)$$

$$\mu = p_{nh} \phi^{nh} \quad (6-b)$$

In this formulation, there are no longer constraint equations. The constraint forces (Lagrange multipliers) are represented by spring and damper forces. No singular positions exist except for the singular configurations of the mass matrix.

$p_h$ ,  $p_{nh}$ ,  $v$  are penalty factors, each one has a physical meaning. For example,  $p_h$  represents the spring stiffness in the constrained direction. In order to avoid constraint violations, the penalty factors must be very large. It leads to numerical convergence problems in the case of implicit numerical procedures and serious instability in the case of explicit numerical procedures.

The penalty formulation can be improved by adding a large artificial mass  $\alpha$  in the constrained direction. Then the system motion is enforced to evolve primarily in the directions with smaller masses (unconstrained direction). The substitute of the reaction forces is made as follows, if we consider the same range of parameters for all the constraints:

$$\lambda = \alpha (p_h \phi^h + v \dot{\phi}^h + \ddot{\phi}^h) \quad (7-a)$$

$$\mu = \alpha (p_{nh} \phi^{nh} + \dot{\phi}^{nh}) \quad (7-b)$$

The governing equations take the form:

$$\mathbf{M}_\alpha \ddot{q} = \mathcal{Q}_T - \alpha \Phi^T \tilde{f}(p_h, v, p_{nh}) \quad (8)$$

with

- $\mathbf{M}_\alpha = \mathbf{M}(q) + \alpha \Phi^T \cdot \Phi$
- $\tilde{f}$  as defined in Eq. (5).

In this full penalty formulation,  $\alpha$  can be considered as a scale factor that can be large. The violation of constraints is reduced but can not be completely eliminated. This may lead to an unstable numerical solution after a long period of simulation.

### 2.4 Acceleration-based augmented formulation

The augmented Lagrangian formulation based on acceleration completes the full penalty formulation by applying an additional constraint reaction force  $\lambda^*$  and  $\mu^*$  to remove the residual motion in the constrained directions at the acceleration level [1,3].

$$\lambda = \lambda^* + \alpha (p_h \phi^h + v \dot{\phi}^h + \ddot{\phi}^h) \quad (9-a)$$

$$\mu = \mu^* + \alpha (p_{nh} \phi^{nh} + \dot{\phi}^{nh}) \quad (9-b)$$

In this formulation, the constraint reaction forces  $\lambda^*$  and  $\mu^*$  are computed iteratively until the generalised acceleration vector  $\ddot{q}$  converges to the solution:

$$\lambda^{(i+1)} = \lambda^{(i)} + \alpha(\ddot{\phi}^h)^{(i+1)} + \alpha(p_h \dot{\phi}^h + v \dot{\phi}^h) \quad (10-a)$$

$$\mu^{(i+1)} = \mu^{(i)} + \alpha(\dot{\phi}^{nh})^{(i+1)} + \alpha p_{nh} \dot{\phi}^{nh} \quad (10-b)$$

with  $\lambda^0 = 0$  and  $\mu^0 = 0$ .

At each iteration,  $q$  and  $\dot{q}$  remain constant. The iteration procedure on the vector of generalised acceleration is defined by:

$$\mathbf{M}(q)\ddot{q}^{(i)} = \mathcal{Q}_T - \Phi_q^{hT} \lambda^{(i)} - \Phi_q^{nhT} \mu^{(i)} \quad (11)$$

$$\mathbf{M}(q)\ddot{q}^{(i+1)} = \mathcal{Q}_T - \Phi_q^{hT} \lambda^{(i+1)} - \Phi_q^{nhT} \mu^{(i+1)}$$

From the definition of  $\mathbf{M}_\alpha$  used in Eq. (8) and of  $\tilde{f}$  in Eq. (5), we have the following expression:

$$\mathbf{M}_\alpha \ddot{q}^{(i+1)} = \mathbf{M}(q)\ddot{q}^{(i)} - \alpha \Phi^T \tilde{f}(p_h, v, p_{nh}) \quad (12)$$

with initial conditions:

$$\ddot{q}^{(0)} = \mathbf{M}(q)^{-1} \mathcal{Q}_T \quad (13)$$

**Remark 1:** The intrinsic advantages of the augmented Lagrangian formulation are due to the inherent features of the leading matrix  $\mathbf{M}_\alpha(q)$ . This matrix remains invertible even in presence of changing topologies (varying constraints), of redundant constraints, singular positions or singular configurations of  $\mathbf{M}(q)$ .

**Remark 2:** The large artificial masses  $\alpha$  are ranging from  $10^5$  to  $10^7$  and few iterations suffice to achieve  $(\ddot{q}^{(i+1)} - \ddot{q}^{(i)}) \leq \varepsilon$ , where  $\varepsilon$  is the required numerical accuracy.

**Remark 3:** The governing equations are established at discrete time  $t_i$ .  $q$  and  $\dot{q}$  are estimated at the same instant by means of the numerical scheme used. Hence, the time step and the stiffness of the constraint must be small enough to obtain:

- small perturbations (time step small enough).
- consistency of the dynamic time response of the reaction force (stiffness small enough).

### 3 NUMERICAL CRITERIA OF STABILITY

#### 3.1 Relative to the penalty parameters

The dynamic behaviour of the penalty function  $\alpha(p_h \dot{\phi}_j^h + v \dot{\phi}_j^h)$  associated to holonomic constraints  $\phi_j^h = 0$  can be approximated by the dynamic behaviour of the following oscillator [2]:

$$\ddot{\phi}_j^h + 2\varepsilon \omega_j \dot{\phi}_j^h + \omega_j^2 \phi_j^h = 0 \quad j \in \{1, 2, \dots, r\} \quad (14)$$

where:

- $\omega_j = \sqrt{\frac{k_j}{m_j}}$  is the natural frequency of the oscillator.
- $\varepsilon = 0.1$  is the numerical dissipation factor.
- $k_j$  is the spring stiffness of the oscillator.
- $m_j$  is the mass in the constrained direction.

This direction is defined by the vector  $\frac{\partial \phi_j^h}{\partial q}$  and  $m_j$  can be calculated as follow:

$$m_j = \frac{\partial \phi_j^h}{\partial q} \mathbf{M}(q) \left( \frac{\partial \phi_j^h}{\partial q} \right)^T \quad (15)$$

Then we can deduce that:

$$\omega_j^2 = \frac{\alpha^h p_j^h}{m_j} \quad (16)$$

From Shannon criteria, we deduce the following conditions:

$$\begin{aligned} \omega_j \ll \frac{2\pi}{\Delta t} &\Rightarrow p_j^h \ll \frac{m_j^h 4\pi^2}{\alpha(\Delta t)^2} \\ &\Rightarrow p_j^h = 0.1 \frac{m_j^h 4\pi^2}{\alpha(\Delta t)^2} \end{aligned} \quad (17)$$

and after identification we obtain:

$$v_j = 2m_j^h \varepsilon \omega_j / \alpha = 2\varepsilon \sqrt{\frac{m_j^h p_j^h}{\alpha}} \quad (18)$$

The dynamic behaviour of the penalty function  $\alpha p_{nh} \dot{\phi}_j^{nh}$  associated to a non-holonomic constraint  $\phi_j^{nh} = 0$  can be approximated by the following dynamic behaviour:

$$\tau_j \dot{\phi}_j^{nh} + \phi_j^{nh} = 0 \quad j \in \{1, 2, \dots, r\} \quad (19)$$

The parameter  $\tau_j$  represents the time constant of the motion (exponential motion) and is given by:

$$\tau_j = \frac{m_j^{nh}}{\alpha p_j^{nh}} \quad (20)$$

where  $m_j^{nh} = \frac{\partial \phi_j^{nh}}{\partial q} M(q) \left( \frac{\partial \phi_j^{nh}}{\partial q} \right)^T$  is the mass of the system in the constrained direction.

The time step  $\Delta t$  must be much smaller than  $\tau_j$  leading to the condition:

$$\begin{aligned} \Delta t \ll \tau_j &\Rightarrow p_j^{nh} \ll \frac{m_j^{nk}}{\alpha \Delta t} \\ &\Rightarrow p_j^{nh} = 0.1 \frac{m_j^{nk}}{\alpha \Delta t} \end{aligned} \quad (21)$$

### 3.2 Relative to the time increment

It is necessary to have a criteria relative to the time increment in order to control the numerical perturbation at each discrete time. In our formulation this perturbation is represented in Eq. (8) by:

$$\alpha \tilde{f}(p_h, v, p_{nh}) \quad (22)$$

If we project this perturbation in the constrained directions at the acceleration level, we define the following numerical expression:

$$\ddot{\phi} = \alpha \Phi \mathbf{M}_\alpha(q)^{-1} \Phi^T \tilde{f}(p_h, v, p_{nh}) \quad (23)$$

$\ddot{\phi}$  is the vector of constrained acceleration perturbation implicitly depending on the time increment  $\Delta t$ . In explicit numerical scheme the greater is  $\Delta t$ , the greater is  $\ddot{\phi}$  and this leads to numerical instability. The problem is equally encountered in the predictor phase with a predictor/corrector numerical scheme.

Following the standard algorithm, we obtain:

$$\text{If } \ddot{\phi}^T \ddot{\phi} \leq 0.1 \varepsilon_1 \text{ then } \Delta t^* = \frac{3}{2} \Delta t \quad (24\text{-a})$$

$$\text{If } \ddot{\phi}^T \ddot{\phi} > \varepsilon_1 \text{ then } \Delta t^* = \frac{\Delta t}{2} \quad (24\text{-b})$$

where  $\varepsilon_1$  is a numerical tolerance dependent on the numerical scheme and  $\Delta t^*$  is the new time increment.

If the time increment drops, the approximate governing equation at discrete time  $t$  is no longer valid at the acceleration level. Then these equations

must be reformulated at the velocity level (impact equations).

## 4 CHOICE OF GENERALISED COORDINATES

If the set of parameters  $q$  involves only the cartesian coordinates of the absolute position of nodal points, the resulting mass matrix is constant. For these so called ‘‘natural coordinates’’, no centrifugal or Coriolis forces appear in the motion equations. However, in rigid bodies dynamics, these natural coordinates are not independent and a set of algebraic constraints must be added [10,11].

Another way to model the connected rigid bodies is the very efficient reduced-coordinates (joint coordinates) formulation. In this case, the coordinates are independent if no kinematic loops exist. However, the mass matrix is time dependent, centrifugal and Coriolis forces as well as singular configurations are present.

In the following, we use natural coordinates as generalised parameters. However, the method described below can be also used with any set of parameters. The most important aspect is the principle of modularity in which connections between bodies can be added or removed during the simulation (contact problems, clearance in the joints, operator decision ...). Following this idea, a multibody system can be considered like a set of components (or substructures) with their own generalised coordinates.

## 5 MODULAR MODEL OF A MULTIBODY SYSTEM

Let us consider a rigid multibody system defined by the following centralized dynamical model :

$$\mathbf{M}_\alpha \ddot{q}^{(i+1)} = \mathbf{M} \ddot{q}^{(i)} - \alpha \Phi^T \tilde{f} \quad (25)$$

with  $\ddot{q}^{(0)} = \mathbf{M}^{-1} Q$ .

Assuming that natural coordinates are used, the mass matrix is constant.

### 5.1 Local equations

Assuming that the multibody system is submitted to  $m$  holonomic constraints and  $r$  non-holonomic one, the first set of constraints involves the internal constraints  $h(q) = 0$ , associated with each rigid body (due to the use of natural coordinates) and the external constraints  $j(q, t) = 0$ , related to the external joints between the bodies. We have the following partitioning of the constraints:

$$\Phi_q^h = \begin{bmatrix} \mathbf{H}_q \\ \mathbf{J}_q \\ \Phi_q^{nh} \end{bmatrix} \quad \text{with} \quad \begin{cases} \mathbf{H}_q = \frac{\partial h}{\partial q} \\ \mathbf{J}_q = \frac{\partial j}{\partial q} \end{cases} \quad \tilde{f} = \begin{Bmatrix} \tilde{f}_H \\ \tilde{f}_J \\ \tilde{f}_\Phi \end{Bmatrix} \quad (26)$$

For a multibody system with  $n$  rigid bodies connected by  $l$  external constraints, the structure of the matrices occurring in the dynamical model is given by:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & & & \\ & \mathbf{M}_2 & & \\ & & \ddots & \\ & & & \mathbf{M}_n \end{bmatrix} \quad \mathbf{H}_q = \begin{bmatrix} \mathbf{H}_{q1}^1 & & & \\ & \mathbf{H}_{q2}^2 & & \\ & & \ddots & \\ & & & \mathbf{H}_{qn}^n \end{bmatrix}$$

$$\mathbf{J}_q = \begin{bmatrix} J_{q1}^1 & J_{q2}^1 & \cdots & J_{qn}^1 \\ J_{q1}^2 & J_{q2}^2 & \cdots & J_{qn}^2 \\ \vdots & \vdots & \ddots & \vdots \\ J_{q1}^l & J_{q2}^l & \cdots & J_{qn}^l \end{bmatrix} \quad \ddot{q} = \begin{Bmatrix} \ddot{q}_1 \\ \vdots \\ \ddot{q}_n \end{Bmatrix} \quad \tilde{f}_H = \begin{Bmatrix} \tilde{f}_{H1} \\ \vdots \\ \tilde{f}_{Hn} \end{Bmatrix}$$

$$\tilde{f}_J = \begin{Bmatrix} \tilde{f}_{J1} \\ \vdots \\ \tilde{f}_{Jl} \end{Bmatrix} \quad \tilde{f}_\Phi = \begin{Bmatrix} \tilde{f}_{\Phi 1} \\ \vdots \\ \tilde{f}_{\Phi r} \end{Bmatrix} \quad Q = \begin{Bmatrix} Q_1 \\ \vdots \\ Q_n \end{Bmatrix}$$

The local equations for a free rigid body  $j$  are deduced:

$$\mathbf{M}_{\alpha j} \ddot{q}_j^{(i+1)} = \mathbf{M}_j \ddot{q}_j^{(i)} - \alpha \mathbf{H}_{qj}^j T \tilde{f}_{Hj} \quad (27)$$

with  $\mathbf{M}_{\alpha j} = \mathbf{M}_j + \alpha \mathbf{H}_{qj}^j T \mathbf{H}_{qj}^j$  and the initial condition  $\ddot{q}_j^{(0)} = \mathbf{M}_j^{-1} Q_j$ .

On the contrary, for a rigid body  $j$  submitted to  $k$  external holonomic constraints and  $p$  external non-holonomic one, the initial conditions are:

$$\ddot{q}_j^{(0)} = \mathbf{M}_j^{-1} (Q_j - F) \quad (28)$$

where:

- $F = \alpha \left( \sum_{k \in [1..l]} J_{qj}^{kT} \hat{f}_{Jk} + \sum_{p \in [1..r]} \Phi_{qj}^{pT} \hat{f}_{\Phi p} \right)$
- $\hat{f}_J = f_J - \tilde{f}_J$
- $\hat{f}_\Phi = f_\Phi - \tilde{f}_\Phi$

We can notice that all the external constraints of the body  $j$  (holonomic and non-holonomic) are taken into account only in the first iteration, corresponding thus to a simple penalty formulation. In the following iterations, only internal constraints of the body  $j$  are taken into account corresponding to an acceleration-based augmented Lagrangian formulation.

Each body can be considered as a numerical component with its own:

- Mechanical parameters (mass matrix, stiffness matrix, damping matrix).
- Numerical parameters  $p_h$  associated to internal holonomic constraints.
- Numerical integrator scheme.

It is necessary to consider another numerical component in which violations of external constraints between rigid bodies are calculated at any time. It is the joint component which communicates to the joined bodies the corresponding value inside the vector  $F$ .

A modular model is well suited to object-oriented programming (OOP) where each body and each joint can be modelled separately. It extends the concept of object-oriented graphic design, already used in domains such as animation, to the concept of object-oriented numerical design. Readers can have more details on these important aspects of OOP in a recent publication by the authors [7].

## 5.2 Algorithm

With the above considerations the general algorithm is:

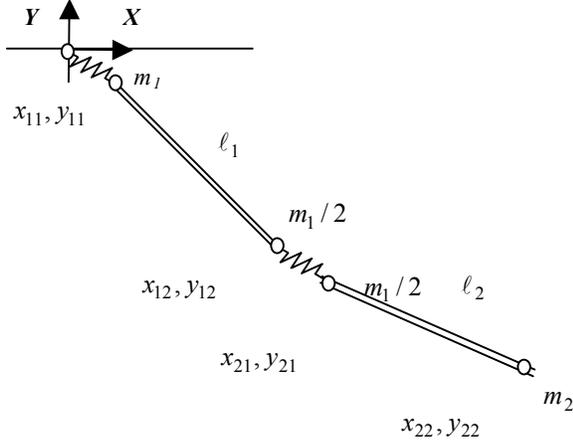
At the discrete time  $t_i$  is given  $\Delta t$ ,  $q$ ,  $\dot{q}$ .

1. Each joint component gives to the connected bodies the corresponding value inside the vector  $F$ , after having calculated the correct penalty parameters. These parameters must conform to the actual time increment  $\Delta t$  and the fixed parameter  $\alpha_i$  of each body  $i$ , Eq. (17), (18) and (21).
2. Each body component  $i$  computes its own generalized acceleration vector according to the iteration procedure defined in Eq. (25).
3. Each body component calls its own numerical integration subroutine to compute  $q_i$ ,  $\dot{q}_i$  at time step  $t + \Delta t$  and communicates these state variables to the joint components to which it is connected.
4. Go to step  $t_{i+1} = t_i + \Delta t$

## 6 NUMERICAL RESULTS

### 6.1 Double pendulum

The following results in Fig 2, Fig. 3 and Fig. 4 represent the comparison between the simulation of the centralized dynamic model and the modular dynamic model of a double pendulum in fully cartesian coordinates (Fig. 1).



**Fig. 1: Double pendulum in fully cartesian coordinates**

The dynamic model has 8 degrees of freedom constrained by 6 holonomic constraints defined by:

$$q^T = \{x_{11} \ y_{11} \ x_{12} \ y_{12} \ x_{21} \ y_{21} \ x_{22} \ y_{22}\} \quad (29)$$

$$h = \begin{cases} (x_{11} - x_{12})^2 + (y_{11} - y_{12})^2 - l_1^2 \\ (x_{21} - x_{22})^2 + (y_{21} - y_{22})^2 - l_2^2 \end{cases} \quad j = \begin{cases} x_{11} \\ y_{11} \\ x_{12} - x_{21} \\ y_{12} - y_{21} \end{cases} \quad (30)$$

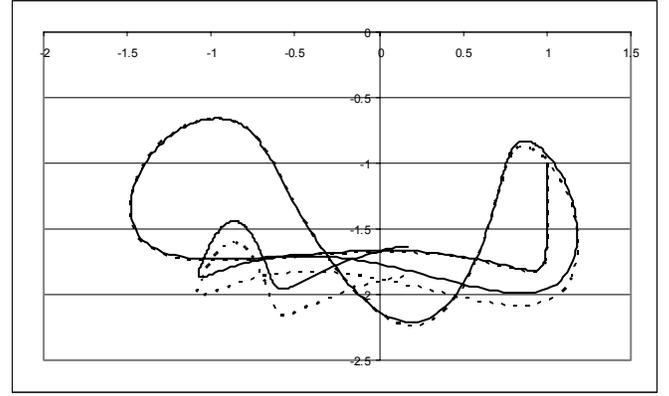
where:

- $l_1 = 1$                        $l_2 = 1$
- $m_1 = 1$                        $m_2 = 1$

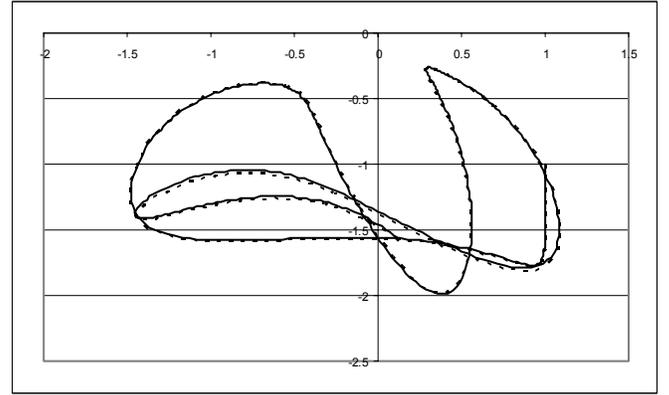
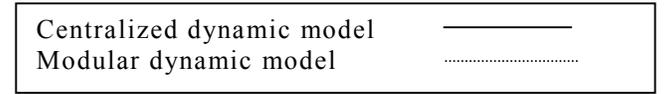
and the initial conditions are:

- $q_0^T = \{0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ -1\}$
- $\dot{q}_0 = \{0\}$

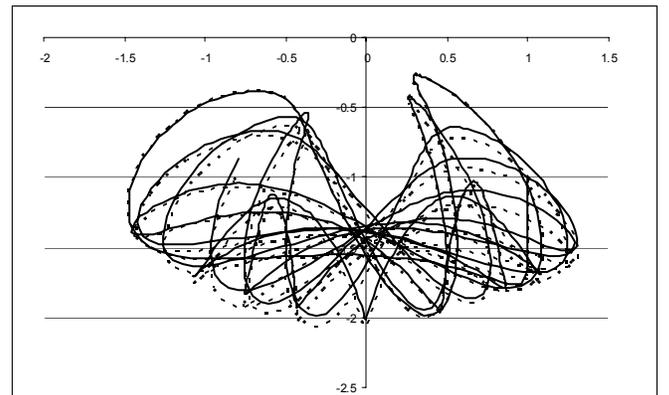
Figure 4 shows very good numerical stability of our algorithm with an increment time step of  $10^{-3}$  and the results are similar with the centralized dynamic model.



**Fig. 2: Trajectory of the mass point  $m_2$   
 $\Delta t = 10^{-2}$ , 5s**



**Fig. 3: Trajectory of the mass point  $m_2$   
 $\Delta t = 10^{-3}$ , 5s**



**Fig. 4: Trajectory of the mass point  $m_2$   
 $\Delta t = 10^{-3}$ , 20s**

With a larger time step of  $10^{-2}$  the results obtained from the two models are not the same

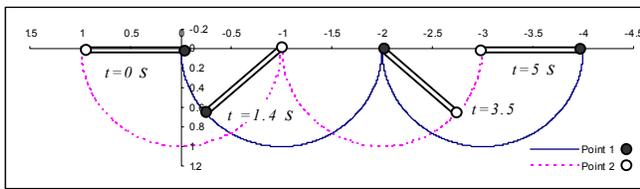
(Fig.2), because the spring stiffness of the external link has been divided by 100 for the sake of numerical stability.

Figure 3 shows more similar results between the two models than in Fig. 2 because the spring stiffness is large enough to verify approximately the geometrical external constraints.

## 6.2 Simple pendulum with changing topology

The second example is a simple pendulum composed of a rigid link of mass  $m$  and length  $\ell$ . The four parameters  $(x_1, y_1, x_2, y_2)$  used to represent the system are the absolute cartesian coordinates of points 1 and 2 (respectively black and white points on Fig. 5):

$$q^T = \{x_1 \ y_1 \ x_2 \ y_2\} \quad (31)$$



**Fig. 5: Trajectory of a double pendulum with changing topology**

The initial conditions are defined as follow:

- $q_0^T = \{0 \ 0 \ -1 \ 0\}$  and  $\dot{q}_0 = \{0\}$
- Point 1 is fixed to the ground
- Point 2 is free

The system is also subject to varying constraints: each time the free end meets x-axis, the constrained end becomes free and vice-versa.

This one degree of freedom system needs three constraints composed of:

- two external constraints to express the joint
- one internal constraint defining the rigid link and defined by:

$$h = \{(x_1 - x_2)^2 + (y_1 - y_2)^2 - \ell^2\} \quad j = \begin{Bmatrix} x_1 \\ y_1 \end{Bmatrix} \quad (32)$$

The results shown in Fig. 5 point out the efficiency of the method for a system with changing topology. The same algorithm is able to take into account the creation or the deletion of constraints with no additional transformation.

## 7 CONCLUSION

In this paper, a modular model of multibody systems using the acceleration-based augmented Lagrangian formulation is presented. This work is the first step to validate the concept of numerical components, involving:

- Numerical Joint
- Numerical solid component

The results presented in this paper are obtained by using the same numerical scheme (Newmark  $\beta$ -method with  $\alpha=1/2$  and  $\beta=1/4$ ) for the two components of the multibody system; moreover, we assume that the increment time step is constant all along the simulation. In future works, more complex systems will be investigated, in order to show the main interest of this modular method, which is that each component of the system can be designed by its own kinematical model, its own numerical scheme and its own time step. More complex systems can be investigated, for example flexible multibody systems with changing topology can be solved.

Thus we can imagine the future work of the mechanical designer will consist in rapidly and safely connecting numerical components already tested by other mechanical designers.

## REFERENCES

- [1] Bayo E., Garcia de Jalon J., and M. A. Serna M. A., 1988, "A Modified lagrangian formulation for the dynamics analysis of constrained mechanical systems". *Computer methods in applied mechanics and engineering*, pp. 183-195.
- [2] Baumgarte J. W., 1972, "Stabilization of constraints and integrals of motion in dynamical", *Computer Method in Applied Mechanics and Engineering*, 1, pp. 1-16.
- [3] Blajer W., 2002, "Augmented Lagrangian Formulation: Geometrical Interpretation and Application to Systems with Singularities and Redundancy", *Multibody System Dynamics* 8, pp. 141-159.
- [4] Cardona A. and Géradin M., 1991, "Modelling of superelements in mechanism analysis", *Int. Jour. Num. Meth. Engineering*, Vol. 32, No 8, pp. 1565-1594.
- [5] Chiou J. C., Park K. C., Farhat C., 1993, "A natural partitioning scheme for parallel simulation of multibody systems", *Int. Jour. Num. Meth. Engineering*, Vol. 36, pp. 945-967.

- [6] Dixit D. S., Shanbhag S. H., Mudur S. P., Isaac K., Chinchalkar S., 1999, "Object-oriented design of an interactive mechanism simulation system-Clodion", *Computer and Graphics*, 23, pp. 85-94.
- [7] Feng Z.-Q, Joli P., Séguy N., 2003, "FER/Mech - A software with interactive graphics for dynamic analysis of multibody system", To appear in *Advances in Engineering Software*.
- [8] Gérardin M., 1993, "Aspects numériques et logiciels de l'approche éléments finis à l'analyse dynamique des systèmes articulés flexibles", *Coll. Nat. en calcul des structures*, 11-14 Mai, Giens, pp. 13-32.
- [9] Hairer E., Lubich C., Roche M., 1989, "The numerical solution of differential-algebraic systems by Runge-Kutta methods", *Lecture Notes in Mathematics*, Vol. 1409, Springer-Verlag.
- [10] Joli P, Pascal M, Gilbert JR. "Numerical simulation of multibody systems with time dependent structure", 1993, September 19-22, *The 14<sup>th</sup> ASME Biennial Conference on Mechanical Vibration and Noise*, Albuquerque, USA.
- [11] Nikravesh Parvize E. and Affifi Hazem A., 1994, "Construction of the equations of motion for multibody dynamics using point and joint coordinates" *Computer-Aided Analysis of Rigid and Flexible Mechanical System*, pp. 31-60.
- [12] Shabana A. A., 1994, "Computational Dynamics", *John Wiley and Sons*, New York.
- [13] Schiehlen W., "Multibody system Handbook", 1990, *Springer Verlag*, Berlin.
- [14] Tseng F.-C., Ma Z.-D., Hulbert G. M., 2003, "Efficient numerical solution of constrained multibody dynamics systems", *Comput. methods appl. mech. engrg.*, 192, pp. 439-472.