

MUBODYNA

DYNAMIC ANALYSIS

OF MULTIBODY SYSTEMS

A FORTRAN Program
for Dynamic Analysis
of Planar Multibody Systems

User's Manual Version 1.0

By

Paulo Flores



Universidade do Minho
Escola de Engenharia
Dep. Eng. Mecânica

Guimarães - 2010

INDEX

1. INTRODUCTION	01
2. KINEMATIC ANALYSIS OF MULTIBODY SYSTEMS	03
3. DYNAMIC ANALYSIS OF CONSTRAINED MULTIBODY SYSTEMS	05
3.1. Equations of Motions for a Constrained Multibody System	05
3.2. Direct Integration Method of the Equations of Motion	06
3.3. Baumgarte Stabilization Method	08
4. AVAILABLE ELEMENT TYPES	11
5. INPUT FILE	13
(a) General Information	14
(b) Initial Conditions	15
(c) Inertia Properties	15
(d) Constant Applied Forces	15
(e) Ideal Revolute Joints	16
(f) Clearance Revolute Joints	16
(g) Mechanical Properties	17
(h) Ideal Translational Joints	18
(i) Ground Constraints	18
(j) Simple Constraints	18
(k) Driving Constraints	19
(l) Guiding Constraints	19
(m) Translational Spring-Damper-Actuator	20
(n) Rotational Spring-Damper	21
(o) Reaction External Forces	21
(p) Point of Interest	22
(q) Time Parameters	22
6. OUTPUT FILE	24
8. APPLICATION EXAMPLE 2: ½ HAT LEG GUIDANCE MODEL	25
9. REFERENCES	30
10. FLOWCHART OF THE MUBODYNA PROGRAM	31

1. INTRODUCTION

Based on the methodology described in the reference textbook (P.E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988) a computer program designated by MUBODYNA (acronym for Dynamic Analysis of Multibody Systems) was conceived. The program is written in FORTRAN and is organized in a form that can be expanded to include other elements, such as other kinematic joints, etc. This program is based on the Nikravesh's DAP (acronym for Dynamic Analysis Program) with some important modifications. The MUBODYNA code was written by Pedro Moreira, Margarida Machado, Fernando Meireles and Paulo Flores.

The MUBODYNA performs dynamic analysis of two-dimensional mechanical models using multibody systems methodologies, that is, employs the body-coordinate formulation to generate the governing equations of motion for mechanisms. This program also performs dynamic analysis of planar mechanical systems, as original Nikravesh's DAP. The kinematic joints available in the program are revolute joint (ideal and clearance), translational joint and simple kinematic constraints. With these basic joints, a large class of systems can be analysed. It is also possible use in the program driving and guiding constraints.

MUBODYNA is based on the Method of Appended Driving Constraints. Additional constraint equations, called the driving constraints, equal in number of degrees of freedom (DOF) of the system, are appended to the original kinematic constraints. The driving constraints are equations representing each independent coordinate as a function of time. Guiding constraints used in this program can be seen as special type of driving constraints, which impose bodies trajectories based on known center of mass linear and angular positions. MUBODYNA use cubic spline interpolation to compute the guiding constraints data introduced as discrete data.

The Newton-Euler differential equations of motion for each body in the system are employed. For the entire system, the constraint forces are represented in terms of a number of Lagrange multipliers. By appending the kinematic acceleration equations (second time derivative of the kinematic constraints) to the differential equations, the linear system is obtained. LU-factorization is used to solve the linear system for accelerations and Lagrange multipliers. The Runge-Kutta numerical integration algorithm, for instance, then integrates velocities and accelerations to obtain the coordinates and velocities at an updated time.

With this formulation, long time simulations are often subject to constraint violation. This violation, which is a result of accumulated numerical integration errors, becomes more apparent with stiff systems. In order to overcome this difficulty, the Baumgarte Stabilization Method is employed, mainly due to its simplicity and efficiency. The Baumgarte Stabilization Method is very easy to implement and straightforward. Thus, changes have been made to this program to accommodate for the stabilizing values α and β , accordingly to Baumgarte Stabilization Method.

Revolute joints with clearance (with dry contact and lubrication effect) are also possible to include in the MUBODYNA program. If clearance is allowed in a revolute joint, the two kinematic constraints are removed and two degrees of freedom are introduced instead. Though a clearance joint does not constrain any degrees of freedom of the mechanical systems, it imposes some boundary conditions, limiting the journal to move within the bearing.

When contact between the journal and the bearing is detected, the contact forces are calculated, accordingly to a constitutive law, and included in the equations of motion for the multibody system.

The program can model constant forces, gravity, translational elements consisting of a spring, a damper, and/or an actuator, and torsional elements consisting of a spring or a damper. MUBODYNA provides also the evaluation of the reaction external forces acting on the system by using a cubic spline interpolation to compute the reaction external forces data introduced as discrete data.

If the forces acting on a mechanical system are known, then the equations of motion can be solved to obtain the motion of the system. This process is known as forward dynamic analysis and it's what MUBODYNA program performs. In some problems, a specified motion for a mechanical system is sought and the objective is to determine the forces that must act on the system to produce such a motion. This process is usually referred to as inverse dynamic or kinetostatic analysis.

The user is free to use any desired unit system. However, it is user's responsibility to enter all the input data in a consistent system.

2. KINEMATIC ANALYSIS OF MULTIBODY SYSTEMS

The kinematic analysis is the study of the motion of a system, independently of the forces that produce it. This type of analysis is performed by solving a set of equations that result from the kinematic, driving and guiding constraint setup.

When the configuration of a multibody system is described by n Cartesian coordinates, then a set of m algebraic kinematic independent holonomic constraints Φ can be written in a compact form as (Nikravesh, 1988),

$$\Phi(\mathbf{q}, t) = \mathbf{0} \quad (1)$$

where \mathbf{q} is the vector of generalized coordinates and t is the time variable, in general associated with the driving and guiding elements.

The velocities and accelerations of the system elements are evaluated using the velocity and acceleration constraint equations,

$$\Phi_{\mathbf{q}} \dot{\mathbf{q}} = -\Phi_t \equiv \mathbf{v} \quad (2)$$

$$\Phi_{\mathbf{q}} \ddot{\mathbf{q}} = -(\Phi_{\mathbf{q}} \dot{\mathbf{q}})_{\mathbf{q}} \dot{\mathbf{q}} - 2\Phi_{\mathbf{qt}} \dot{\mathbf{q}} - \Phi_{tt} \equiv \boldsymbol{\gamma} \quad (3)$$

where $\Phi_{\mathbf{q}} = \partial \Phi / \partial \mathbf{q}$ is the Jacobian matrix of the constraint equations, $\dot{\mathbf{q}}$ is the vector of generalized velocities, $\mathbf{v} = \partial \Phi / \partial t$ is the right hand side of velocity equations, $\ddot{\mathbf{q}}$ is the acceleration vector and $\boldsymbol{\gamma}$ is the right hand side of acceleration equations, *i.e.*, the vector of quadratic velocity terms, which contains the terms that are exclusively function of velocity, position and time.

It should be noticed that only rheonomic constraints, associated with driving and guiding equations, contribute with non-zero entries to the vector \mathbf{v} . Furthermore, it is assumed that this vector does not present any dependency on the vector of coordinates.

The kinematic analysis of a multibody system can be carried out by solving the set of Equations (1), (2) and (3). The necessary steps to perform this analysis are summarized as sketched in Figure 1, and described as,

- i) Specify the initial conditions for positions \mathbf{q}^0 and initialize the time counter t^0 .
- ii) Evaluate the position constraint equations (1) and solve them for positions, \mathbf{q} .
- iii) Evaluate the velocity constraint equations (2) and solve them for velocities, $\dot{\mathbf{q}}$.
- iv) Evaluate the acceleration constraint equations (3) and solve them for accelerations, $\ddot{\mathbf{q}}$.
- v) Increment the time. If the time is smaller than final time, go to step ii), otherwise stop the kinematic analysis.

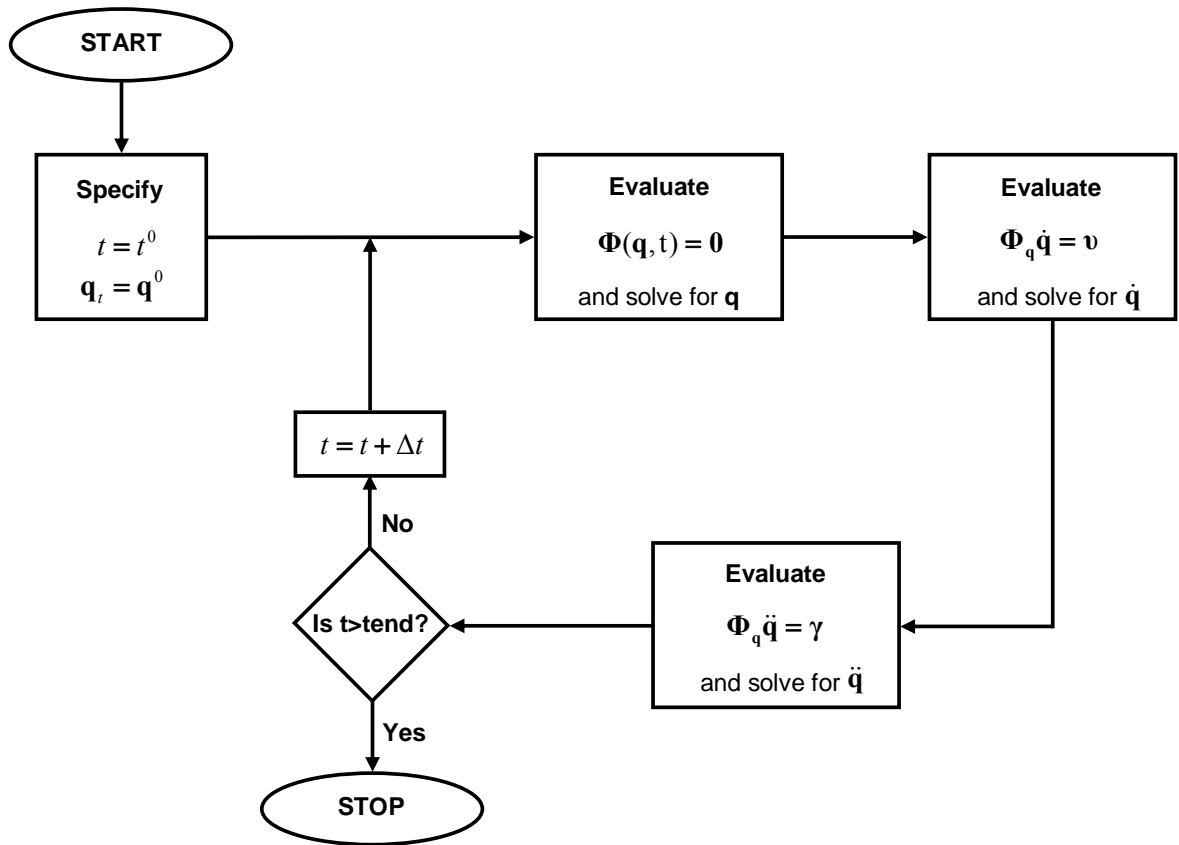


Figure 1: Flowchart of computational procedure for kinematic analysis of a multibody system.

3. DYNAMIC ANALYSIS OF CONSTRAINED MULTIBODY SYSTEMS

The dynamic analysis of multibody systems aims to understanding the relationship between the motion of the system parts and the causes that produce the motion including external applied forces and moments. The motion of the system is, in general, not prescribed, being its calculation one of the principal objectives of the analysis. The dynamic analysis also provides a way to estimate external forces that depend on the relative position between the system's components, such as the forces exerted by springs, dampers and actuators. Furthermore, it also possible to estimate the external forces that are developed as a consequence of the interaction between the system components and the surrounding environment, such as contact-impact forces, friction forces and external reaction forces. The internal reaction forces and moments generated at the kinematic joints are also provided by the dynamic analysis. These reaction forces and moments prevent the occurrence of the relative motions in the prescribed directions between the bodies connected via kinematic joints.

3.1. Equations of Motions for a Constrained Multibody System

The equations of motion for a constrained multibody system of rigid bodies are written as (Nikravesh, 1988),

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{g} + \mathbf{g}^{(c)} \quad (4)$$

where \mathbf{M} is the system mass matrix, $\ddot{\mathbf{q}}$ is the vector that contains the system accelerations, \mathbf{g} is the generalized force vector, which contains all external forces and moments, and $\mathbf{g}^{(c)}$ is the vector of constraint reaction equations. The joint reaction forces can be expressed in terms of the Jacobian matrix of the constraint equations and the vector of Lagrange multipliers as (Nikravesh, 1988),

$$\mathbf{g}^{(c)} = -\Phi_q^T \boldsymbol{\lambda} \quad (5)$$

where $\boldsymbol{\lambda}$ is the vector that contains m unknown Lagrange multipliers associated with m holonomic constraints. The Lagrange multipliers are physically related to the reaction forces and moments generated between the bodies interconnected by kinematic joints. Thus, substitution of Equation (5) in Equation (4) yields,

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_q^T \boldsymbol{\lambda} = \mathbf{g} \quad (6)$$

In dynamic analysis, a unique solution is obtained when the constraint equations are considered simultaneously with the differential equations of motion, for a proper set of initial conditions. Therefore, Equation (3) is appended to Equation (6), yielding a system of differential algebraic equations. Thus, mathematically the simulation of constrained multibody system requires the solution of a set of nc differential equations coupled with a set of m algebraic equation, written as,

$$\begin{bmatrix} \mathbf{M} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{g} \\ \boldsymbol{\gamma} \end{Bmatrix} \quad (7)$$

Then, in each integration time step, the accelerations vector, $\ddot{\mathbf{q}}$, together with velocities vector, $\dot{\mathbf{q}}$, are integrated in order to obtain the system velocities and positions for the next time step. This procedure is repeated up to final time analysis time is reached.

The system of the motion equations (7) does not use explicitly the position and velocity equations associated with the kinematic constraints, that is, Equations (1) and (2). Consequently, for moderate or long simulations, the original constraint equations start to be violated due to the integration process and/or to inaccurate initial conditions. Therefore, special procedures must be followed to avoid or minimize this phenomenon. Several methods to solve this problem have been suggested and tested, being the most common among them the Baumgarte Stabilization Method (Baumgarte, 1972), the Coordinate Partitioning Method (Wehage and Haug, 1982) and the Augmented Lagrangian Formulation (Bayo *et al.*, 1988).

3.2. Direct Integration Method of the Equations of Motion

In order to advance the analysis in time, the equations of motion need to be integrated. Considering the high level of complexity of these equations, analytical solutions are impractical to obtain and, consequently, numerical integration algorithms must be applied.

The standard integration of the equations of motion, here called Direct Integration Method, converts the nc second-order differential equations of motion into $2nc$ first-order differential equations. Then, a numerical scheme, such as the fourth-order Runge-Kutta method, is employed to solve the initial-value problem (Shampine and Gordon, 1975; Gear, 1981).

The commonly used numerical integration algorithms are useful in solving first-order differential equations that take the form (Shampine and Gordon, 1975),

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, t) \quad (8)$$

Thus, if there are nc second-order differential equations, they are converted to $2nc$ first-order equations by defining the \mathbf{y} and $\dot{\mathbf{y}}$ vectors, which contains, respectively, the system positions and velocities and the system velocities and accelerations, as follows,

$$\mathbf{y} = \begin{Bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{Bmatrix} \quad \text{and} \quad \dot{\mathbf{y}} = \begin{Bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{Bmatrix} \quad (9)$$

The reason for introducing the new vectors \mathbf{y} and $\dot{\mathbf{y}}$ is that most numerical integration algorithms deal with first-order differential equations (Shampine and Gordon, 1975). The following diagram can interpret the process of numerical integration at instant of time t ,

$$\dot{\mathbf{y}}(t) \xrightarrow{\text{Integration}} \mathbf{y}(t + \Delta t) \quad (10)$$

velocities and accelerations at instant t , after integration process, yield positions and velocities at next time step, $t=t+\Delta t$.

Figure 2 presents a flowchart that shows the algorithm of Direct Integration Method of the equations of motion. At $t=t^0$, the initial conditions on \mathbf{q}^0 and $\dot{\mathbf{q}}^0$ are required to start the integration process. These values cannot be specified arbitrarily, but must satisfy the constraint equations defined by Equations (1) and (2).

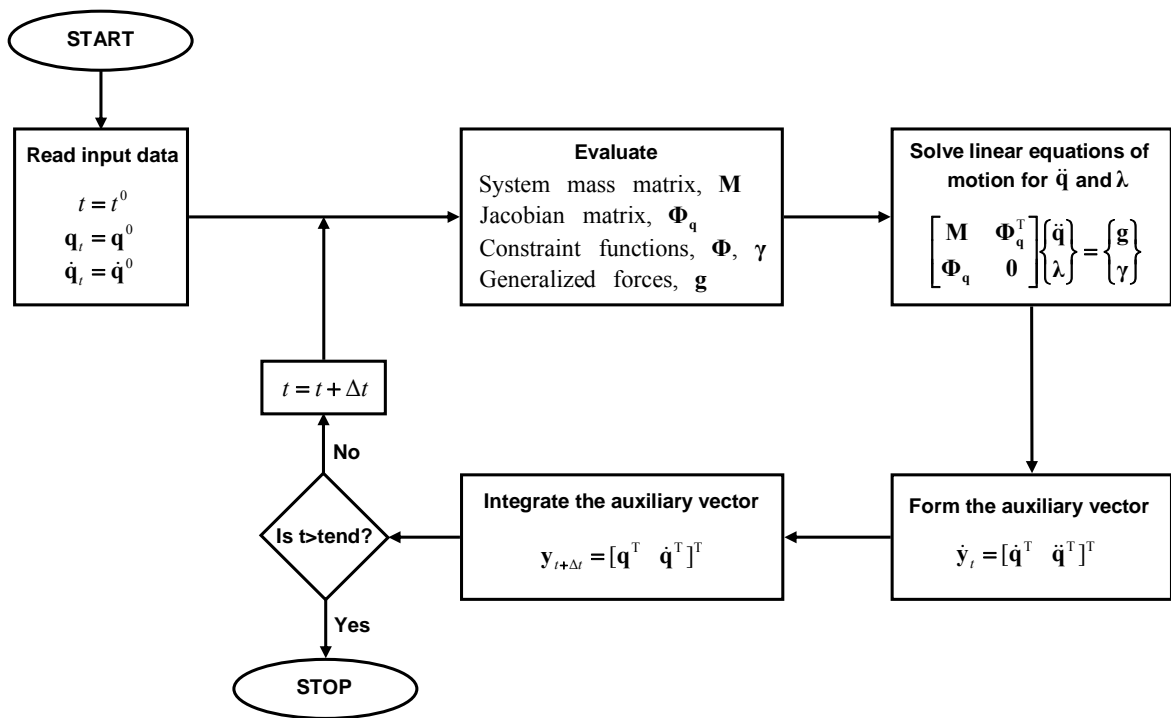


Figure 2: Flowchart of computational procedure for dynamic analysis of multibody systems - Direct Integration Method.

The direct integration algorithm presented in Figure 2 can be summarized by the following steps:

- i) Start at instant of time t^0 with given initial conditions for positions \mathbf{q}^0 and velocities $\dot{\mathbf{q}}^0$.
- ii) Assemble the global mass matrix \mathbf{M} , evaluate the Jacobian matrix Φ_q , construct the constraint equations Φ , determine the right hand side of the accelerations γ , and calculate the force vector \mathbf{g} .
- iii) Solve the linear set of the equations of motion (7) for a constrained multibody system in order to obtain the accelerations $\ddot{\mathbf{q}}$ at time t and the Lagrange multipliers λ .

- iv) Assemble the vector $\dot{\mathbf{y}}_t$ containing the generalized velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$ for instant of time t .
- v) Integrate numerically the $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ for time step $t+\Delta t$ and obtain the new positions and velocities.
- vi) Update the time variable and go to step ii) and proceed with the process for the new time step. Perform these steps until the final time of analysis is reached.

The Direct Integration Method of equations of motion is highly prone to integration errors, because the constraint equations (1) and (2) are only satisfied at the initial instant of time. In the first few time steps, the constraint violations are usually small and negligible. However, as time progresses, the error in computed values for kinematic parameters is accumulated and constraint violations increase. Hence, the results produced can be unacceptable; therefore, the Direct Integration Method requires constraint stabilization, especially for long simulations. It should be noted that the Direct Integration Method is quite sensitive to initial conditions, which can be an important source of errors in the integration process.

3.3. Baumgarte Stabilization Method

Due to its simplicity and easiness for computational implementation, the Baumgarte Stabilization Method is the most popular and attractive technique to overcome the drawbacks of the standard integration of the equations of motion. Baumgarte's method can be looked upon as an extension of feedback control theory. The principle of this method is to damp out the acceleration constraint violations by feeding back the violations of the position and velocity constraints. The choice of the feedback parameters depends on several factors, namely, the integrator used and the model of the multibody system. This parameters' choice is the major drawback of Baumgarte's method.

As shown before, for a multibody system of kinematically constrained rigid bodies, the equations of motion are given by,

$$\begin{bmatrix} \mathbf{M} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{g} \\ \gamma \end{Bmatrix} \quad (11)$$

with initial conditions for \mathbf{q}^0 and $\dot{\mathbf{q}}^0$. The constraint violations result from accumulated numerical integration errors, and become more apparent with stiff systems, that is, when the natural frequencies of the system are widely spread. The stiffness can be produced by the physical characteristics of the multibody system, such as components with large differences in their masses, stiffness and/or damping (Garcia de Jalon and Bayo, 1994). Even though the initial conditions guarantee the non-violation of constraint equations on position (1) and velocity level (2), during the course of numerical integration the numerical errors do not satisfy the constraint equations, and the effect of these errors increases with time.

When Equation (11) is solved, actually Equations (3) and (4) are not considered. In fact, only the second derivatives of constraint equations will be satisfied every integration step. However, it is known that Equation (3) represents an unstable system (Cochin and Cadwallender, 1997; Franklin *et al.*, 2002). The Baumgarte Stabilization Method allows constraints to be slightly violated before corrective actions can take place, in order to force the violation to vanish.

The goal of the Baumgarte Stabilization Method is to replace the differential Equation (3) by the following equation (Baumgarte, 1972),

$$\ddot{\Phi} + 2\alpha\dot{\Phi} + \beta^2\Phi = 0 \quad (12)$$

Equation (12) is the differential equation for a closed-loop system in terms of kinematic constraint equations. The terms $2\alpha\dot{\Phi}$ and $\beta^2\Phi$ in Equation (12) play the role of a control terms. The principle of the method is based on the damping of acceleration of constraint violation by feeding back the position and velocity of constraint violations, as seen in Figure 3, which shows open-loop and closed-loop control systems. In the open-loop systems Φ and $\dot{\Phi}$ will not converge to zero if any deviation from zero occurs and, therefore, the system is unstable.

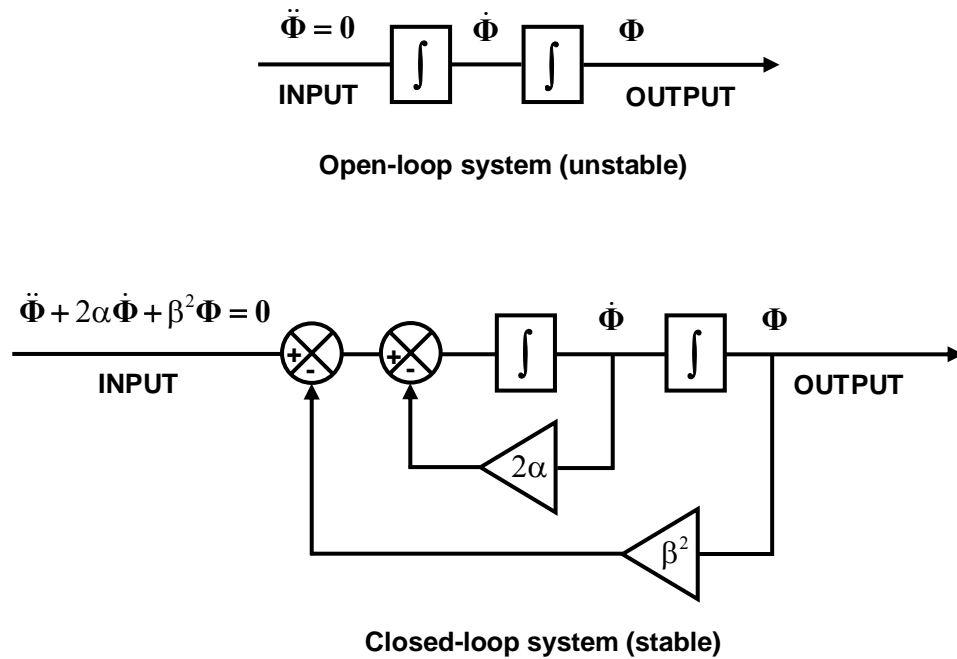


Figure 3: Open-loop (cascade) and closed-loop (feedback) control systems.

Thus, utilizing the Baumgarte's approach, the equations of motion for a dynamic system subjected to holonomic constraints are stated in the form,

$$\begin{bmatrix} \mathbf{M} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{g} \\ \gamma - 2\alpha\dot{\Phi} - \beta^2\Phi \end{Bmatrix} \quad (13)$$

In general, if α and β are chosen as positive constants, the stability of the general solution of Equation (2.18) is guaranteed. When α is equal to β , critical damping is achieved, which usually stabilizes the system response more quickly. Baumgarte (1972) highlighted that the suitable choice of the parameters α and β is performed by numerical experiments. Consequently, the Baumgarte's method still has some ambiguity in determining optimal feedback gains. Indeed, it seems that the value of the parameters is purely empiric, and there is no reliable method for selecting the coefficients α and β . The improper choice of these coefficients can lead to unacceptable results in the dynamic simulation of multibody mechanical systems.

More recently, some works have suggested criteria that help in the selection of the parameters α and β . Chang and Nikravesh (1985) presented an adaptive numerical scheme for determining these parameters. The stability requirement of the adaptation mechanism proposed by Chang and Nikravesh, is based on the critical damping, that is, $\alpha=\beta$. The idea of this approach is based on the adaptive scheme in which the feedback parameters are adjusted according to the predicted error. However, this approach requires complex programming and increases computational time. Later, Kim *et al.* (1990) presented a method for determining the feedback parameters. This method is based on the stability analysis for the error propagation of the modified constrained equation containing the feedback parameters, which is discretized by the Runge-Kutta algorithm in a time domain. Lin and Hong (1998) showed that the parameters α and β can depend on the integration method. In their study, the stability analysis methods used in digital control theory are applied to solve the instability problem.

Baumgarte (1972) pointed out that the stabilizing values of $\alpha=\beta=5$ is a good choice for a MBS made of rigid bodies, still, there is no clear-cut method for choosing the most correct values for feedback parameters for general cases. The choice of the α and β parameters usually involves a trial-and-error-procedure. Furthermore, the selection of the stabilization parameters depends on the type of application, namely, the integrator, the integration step size and the type of model of multibody system. However, the parameters α and β should be, in general, equal one another and typical values of the stabilization parameters range from 1 to 20 (Nikravesh, 1988; Garcia de Jalon and Bayo, 1994).

4. AVAILABLE ELEMENT TYPES

A typical multibody system can be defined as a collection of rigid or flexible bodies interconnected by kinematic joints and possibly some force elements. The kinematic joints control the relative motion between the bodies, while the force elements represent the internal forces that develop between bodies due to their relative motion. The forces applied over the system components may be the result of springs, dampers, actuators or external forces. A generic description of a general multibody model is represented in Figure 4.

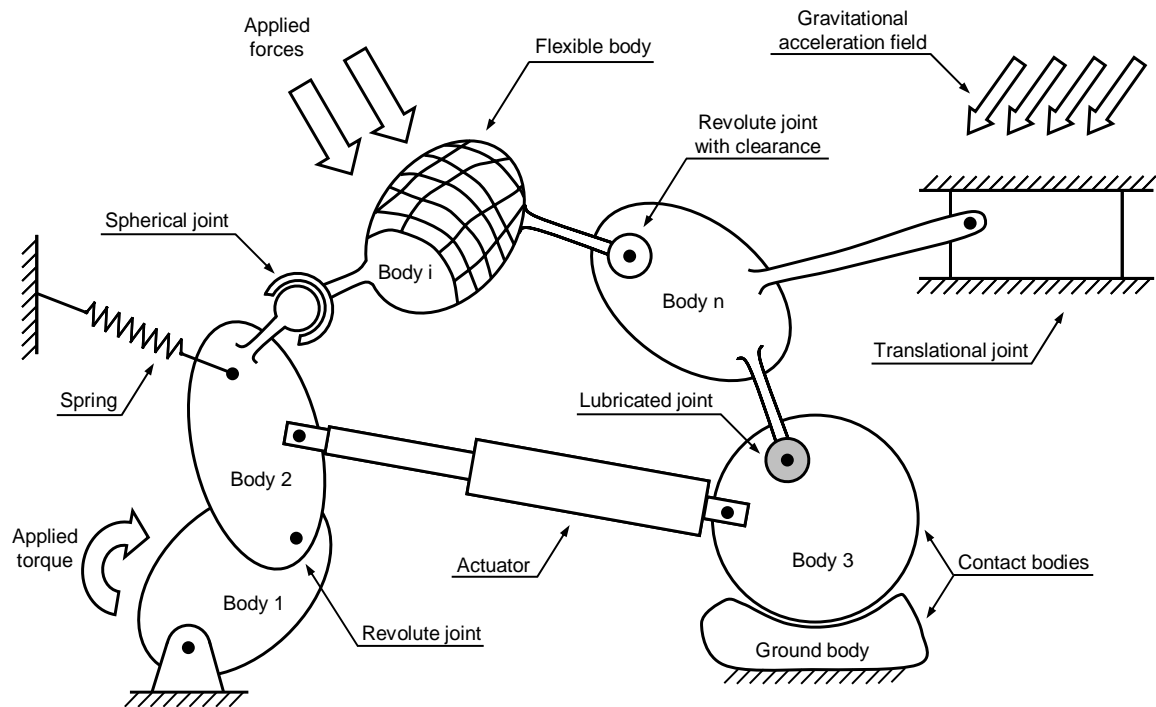


Figure 4: Schematic representation of a general multibody system. The most significant elements are described as rigid bodies interconnected by kinematic joints. External applied forces of different levels of complexity act on the system in order to simulate the interactions among the elements and between these and the surrounding environment. Driving and guiding elements can also be represented under this definition of multibody system.

MUBODYNA is based on a formulation that only rigid bodies are considered. This program has also a restricted type of kinematic and force elements available.

The available element types in the MUBODYNA are the following:

- Rigid bodies
- Kinematic Constraints:
 - Ideal revolute joints
 - Ideal translational joints
 - Ground constraints
 - Simple constraints

- Driving constraints
- Guiding constraints
- Force elements:
 - Revolute joints with clearance (dry contact – with and without friction)
 - Revolute joints with clearance (hydrodynamic lubrication effect)
 - Translational spring-damper-actuator
 - Rotational spring-damper
 - Constant force and moment
 - Reaction External Forces

5. INPUT FILE

A multibody must be provided to MUBODYNA as an input file. The input file is the file that contains the model description, and must be a text file. The elements of a multibody system are described in data lines.

A list of all the prompts given by the program MUBODYNA is presented below. The prompts are labeled for easy reference, from (a) to through (q). The prompts given are as follows:

- (a) **GENERIC INFORMATION** – Number of multibody system elements:
- | | |
|------|--|
| NB | Number of rigid bodies |
| NIRJ | Number of ideal revolute joints |
| NCRJ | Number of clearance revolute joints |
| NTJ | Number of translational joints |
| NGB | Number of ground bodies |
| NSC | Number of simple constraints |
| NDC | Number of driving constraints |
| NGC | Number of guiding constraints |
| NGD | Number of guiding data or points associated with guiding constraints |
| NTS | Number of translational spring-damper-actuators elements |
| NRS | Number of rotational spring-damper elements |
| NBEF | Number of bodies subjected to external forces |
| NFD | Number of forces/data points for external applied forces |
| NPI | Number of points of interest |
- (b) **INITIAL CONDITIONS** – Global Cartesian coordinates on the positions and velocities:
INITIAL CONDITIONS ON X, Y, PHI; INITIAL CONDITIONS ON XD, YD, PHID.
- (c) **INERTIA PROPERTIES** – Mass and moment of inertia of the bodies:
MASS m and MOMENT OF INERTIA I .
- (d) **CONSTANT APPLIED FORCES** – Constant forces and moments acting on bodies:
FORCE COMPONENT X F_x , FORCE COMPONENT Y F_y and MOMENT M .
- (e) **IDEAL REVOLUTE JOINTS** – Body numbers i and j and local coordinates of the points P_i and P_j :
BODY NOS. I AND J; LOCAL COORDINATES XI-P-I; ETA-P-I; XI-P-J; ETA-P-J.
- (f) **CLEARANCE REVOLUTE JOINTS** – Body numbers i and j , local coordinates of the points and radius R_i and R_j :
BODY NOS. I AND J; LOCAL COORDINATES XI-P-I; ETA-P-I; XI-P-J; ETA-P-J; RADIUS R_i and R_j .
- (g) **MECHANICAL PROPERTIES** – Physical characteristics of the clearance joints:
FORCE LAW; BEARING FRICTION COEFFICIENT $FRIC_i$; BEARING RESTITUTION COEFFICIENT $REST_i$; BEARING YOUNG'S MODULE E_i ; BEARING POISSON'S RATIO NIU_c ; JOURNAL FRICTION COEFFICIENT $FRIC_j$; JOURNAL RESTITUTION COEFFICIENT $REST_j$; JOURNAL YOUNG'S MODULE E_j ; JOURNAL POISSON'S RATIO NIU_j ; DYNAMIC FLUID VISCOSITY;
- (h) **TRANSLATIONAL JOINTS** - Body numbers i and j and local coordinates of the points P_i , Q_i and P_j :
BODY NOS. I AND J; LOCAL COORDINATES XI-P-I; ETA-P-I; XI-Q-I; ETA-Q-I; XI-P-J; ETA-P-J.
- (i) **GROUND CONSTRAINTS** – Body(ies) number(s) constrained to the ground:
BODY NO. I.
- (j) **SIMPLE CONSTRAINTS** – Simple constraints:
BODY NO. 1, 2, or 3 for X, Y OR PHI CONSTRAINT DIRECTION.
- (k) **DRIVING CONSTRAINTS** – Driving constraints:
BODY NO. 1, 2, or 3 for X, Y OR PHI CONSTRAINT DIRECTION; INITIAL POSITION;
INITIALVELOCITY; INITIAL ACCELERATION.
- (l) **GUIDING CONSTRAINTS** – Guiding constraints:
BODY NO. 1, 2, or 3 for X, Y OR PHI CONSTRAINT DIRECTION; TIME AND GUIDING
CONSTRAINTS DATA.
- (m)

TRANSLATIONAL SPRINGS – Translational spring, damper and actuator:
 BODY NO. I; 1,2 OR 3 FOR X, Y OR PHI CONSTRAINT DIRECTION; XI-P-I; ETA-P-I; XI-P-J; ETA-P-J; SPRING CONSTANT k; DAMPING COEFFICIENT d; UNDEFORMED SPRING LENGTH l.

- (n) **ROTATIONAL SPRINGS** – Rotational spring and damper:
 BODY NO. I; 1,2 OR 3 FOR X, Y OR PHI CONSTRAINT DIRECTION; SPRING CONSTANT k; DAMPING COEFFICIENT d.
- (o) **EXTERNAL APPLIED FORCES** – External applied forces:
 BODY NO I; COORDINATES OF POINT OF APPLICATION XI-P-I and ETA-P-I; APPLIED FORCES Fx and Fy.
- (p) **POINTS OF INTEREST:**
 BODY NO I; XI-P-I; ETA-P-I.
- (q) **TIME PARAMETERS:**
 TIME START; TIME END; TIME STEP; INTEGRATION METHOD; BAUMGARTE PARAMETERS ALPHA AND BETA.

(a) General Information

The first set of data the program requests is

NB	Number of rigid bodies in the system, including ground
NIRJ	Number of ideal revolute joints in the system
NCRJ	Number of clearance revolute joints in the system
NTJ	Number of ideal translational joints in the system
NGB	Number of bodies that are attached to (or considered as) ground
NSC	Number of simple constraints in the system
NDC	Number of driving constraints (driver links)
NGC	Number of guiding constraints
NGD	Number of guiding data or points associated with guiding constraints
NTS	Number of translational spring-damper-actuators elements
NRS	Number of rotational spring-damper elements
NBEF	Number of reference bodies, which have reaction external forces acting on
NFD	Number of input data, associated to guiding constraints, used in the cubic spline interpolation
NPI	Number of points of interest

The program computes the number of coordinates N ($N=3 \times NB$) and the total number of constraint equations M ($M=2 \times NR+2 \times NT+3 \times NG+NS+ND+NGUI$). N must be equal to M. If M is greater than N, then an error message is given. Otherwise, the program continues.

The number of each element tells the program how many data lines should exist. A data line describes the characteristics of each element. The data on a data line can be separated by one or several spaces.

(b) Initial Conditions

The rigid bodies' data line must provide the initial conditions on the positions and velocities for every rigid body in the system.

The initial conditions on the coordinates and velocities must be correct, that is, must be satisfy the position and velocity constraints at the initial time.

The data lines for rigid bodies are repeated NB times, and are as follows:

X Y PHI XD YD PHID

here,

X	<i>X</i> -coordinate of the body
Y	<i>Y</i> -coordinate of the body
PHI	ϕ - coordinate of the body
XD	Velocity of the body in the <i>X</i> -direction
YD	Velocity of the body in the <i>Y</i> -direction
PHID	Rotational velocity of the body

(c) Inertia Properties

The rigid bodies' data line must provide the mass and moment of inertia for every rigid body in the system.

The data lines for revolute joints are repeated NB times, and are as follows:

M INERTIA

where,

M	Mass of body
INERTIA	Moment of inertia of the body

(d) Constant Applied Forces

The rigid bodies' data line must provide the constant external applied forces and/or moments acting at the center of mass for every rigid body in the system.

The data lines for revolute joints are repeated NB times, and are as follows:

FX FY N

where,

FX	Applied force in the X-direction
FY	Applied force in the Y-direction
N	Applied Moment on body

(e) Ideal Revolute Joints

Refer to section 4.2.1 of the textbook (P.E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall, Englewood, New Jersey, 1988), for the detailed description of this ideal kinematic joint.

The data lines for ideal revolute joints are represented NIRJ times, and are as follows:

I	J	XI-P-I	ETA-P-I	XI-P-J	ETA-P-J
----------	----------	---------------	----------------	---------------	----------------

where,

I	Index number of body <i>i</i> (body <i>i</i> have to be the bearing)
J	Index number of body <i>j</i> (body <i>j</i> have to be the journal)
XI-P-I	ξ -coordinate of point <i>P</i> on body <i>i</i>
ETA-P-I	η -coordinate of point <i>P</i> on body <i>i</i>
XI-P-J	ξ -coordinate of point <i>P</i> on body <i>j</i>
ETA-P-J	η -coordinate of point <i>P</i> on body <i>j</i>

(f) Clearance Revolute Joints

Though a clearance joint does not constrain any degrees of freedom of the mechanical systems, it imposes some boundary conditions, limiting the journal to move within the bearing. When contact between the journal and the bearing is detected, the contact forces are calculated, accordingly to a constitutive law, and included in the equations of motion for the multibody system. A revolute joint with clearance is shown, schematically, in figure 5.

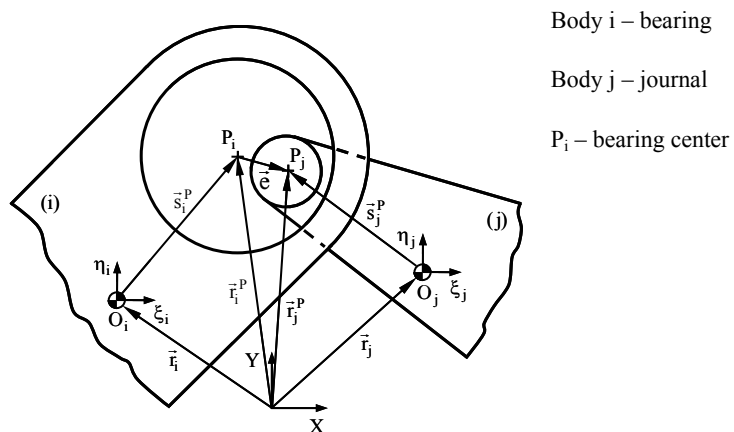


Figure 5: Generic revolute joint with clearance in a multibody system. Clearance is exaggerated for clarity

The data lines for revolute joints with clearance are repeated NCRJ times, and are as follows:

I J XI-P-I ETA-P-I XI-P-J ETA-P-J Ri Rj

where,

I	Index number of body i (body i have to be the bearing)
J	Index number of body j (body j have to be the journal)
XI-P-I	ζ -coordinate of point P on body i
ETA-P-I	η -coordinate of point P on body i
XI-P-J	ζ -coordinate of point P on body j
ETA-P-J	η -coordinate of point P on body j
Ri	Bearing radius
Rj	Journal radius

(g) Mechanical Properties

This is not a typical data line like a joint or a force element. In this data line the program reads material properties of the colliding bodies, that is, the impact between the bearing and the journal in the revolute clearance joints.

The data lines for mechanical properties of clearance joints are repeated NCRJ times, and are as follows:

FORCE LAW FRICi RESTi Ei NIUi FRICj RESTj Ej NIUj VISCOSITY

where,

FORCE LAW	A flag indicating what type of force model is used
FRICi	c_f - Friction coefficient of body i (bearing)
RESTi	c_e - Restitution coefficient of body i (bearing)
Ei	E - Young modulus of body i (bearing)
NIUi	ν - Poisson coefficient of body i (bearing)
FRICj	c_f - Friction coefficient of body j (journal)
RESTj	c_e - Restitution coefficient of body j (journal)
Ej	E - Young modulus of body j (journal)
NIUj	ν - Poisson coefficient of body j (journal)
VISCOSITY	μ - Dynamic viscosity of the fluid used in the lubricated joint

Flag **FORCE LAW** indicates the force model according to the following rule:

EQ.1 : Pinkus and Sternlicht hydrodynamic force model

EQ.2 : Hertz's dry contact force model

EQ.3 : Lankarani & Nikravesh dry contact force model

EQ.4 : Hunt & Crossley dry contact force model

(h) Ideal Translational Joints

Refer to section 4.2.1 of the textbook (P.E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988), for detailed description of this ideal kinematic joint.

The data lines for translational joints are repeated NTJ times, and are as follows:

I J XI-P-I ETA-P-I XI-Q-I ETA-Q-I XI-P-J ETA-P-J

where,

I	Index number of body i
J	Index number of body j
XI-P-I	ξ -coordinate of point P on body i
ETA-P-I	η -coordinate of point P on body i
XI-Q-I	ξ -coordinate of point Q on body i
ETA-Q-I	η -coordinate of point Q on body i
XI-P-J	ξ -coordinate of point P on body j
ETA-P-J	η -coordinate of point P on body j

(i) Ground Constraints

Refer to section 4.2.7 of the textbook (P.E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988), for detailed description of this kinematic constraint.

The data lines for ground constraints are repeated NGB times, and are as follows:

I

where,

I	Index number of body i constrained to the ground
----------	--

(j) Simple Constraints

Refer to section 4.2.7 of the textbook (P.E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988), for detailed description of this kinematic constraint.

The data lines for simple constraints are repeated NSC times, and are as follows:

I CONSTRAINT DIRECTION

where,

I	Index number of body i
CONSTRAINT DIRECTION	Index of the coordinate to be constrained: 1: constrain X -coordinate 2: constrain Y -coordinate 3: constrain ϕ -coordinate

(k) Driving Constraints

Refer to section 4.2.8 of the textbook (P.E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988), for detailed description of this kinematic constraint.

MUBODYNA can provide a driver expression on one of the three coordinates of a body:

1	$x_i = x_i^0 + v_i^0 t + \frac{1}{2} a t^2$
2	$y_i = y_i^0 + v_i^0 t + \frac{1}{2} a t^2$
3	$\phi_i = \phi_i^0 + \omega_i^0 t + \frac{1}{2} \alpha t^2$

The data lines for driving constraints are repeated NDC times, and are as follows:

I CONSTRAINT DIRECTION POSITION VELOCITY ACCELERATION

where,

I	Index number of body i
CONSTRAINT DIRECTION	Index of the coordinate to be constrained: 1: constrain X -coordinate 2: constrain Y -coordinate 3: constrain ϕ -coordinate
POSITION	Initial linear/angular position
VELOCITY	Initial linear/angular velocity
ACCELERATION	Initial linear/angular acceleration

(I) Guiding Constraints

In general the number of guiding constraints must be equal to number of bodies plus 2 ($NGC = NB+2$). This mean that in pure biomechanical systems, the CM (x,y) need to be constrained only for the main body, and ϕ must be constrained for all bodies. A guiding constraint is shown, schematically, in figure 6.

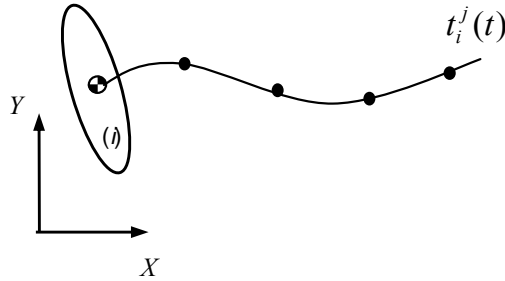


Figure 6: Body trajectory guiding a free body

The data lines for guiding constraints are repeated NGD times, and are as follows:

I CONSTRAINT DIRECTION

where,

I Index number of body i
CONSTRAINT DIRECTION Index of the coordinate to be constrained:
1: constrain X -coordinate
2: constrain Y -coordinate
3: constrain ϕ -coordinate

MUBODYNA provide a guiding expression on one of the three coordinates of a body:

$$1 \quad x_i - t_i^x(t) = 0$$

$$2 \quad y_i - t_i^y(t) = 0$$

$$3 \quad \phi_i - t_i^\phi(t) = 0$$

where $t_i(t)$ is introduced as a set of points as (NGC+1 columns, NGD rows),

Time	Guid_1	Guid_2	Guid_3	Guid_4...	Guid_5	Guid_6
0.0000	0.4727	1.0800	1.4853	1.4434...	0.6946	1.4990
...
0.9570	1.8356	1.0776	1.4476	1.3619	0.7533	1.5803

(m) Translational Spring-Damper-Actuator

Refer to section 9.2.3 to 9.2.5 of the textbook (P.E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988), for detailed description of this force element.

The data lines for translational spring-damper-actuator are repeated NTS times, and are as follows:

I J XI-P-I ETA-P-I XI-P-J ETA-P-J K D FA L0

where,

I	Index number of body i
J	Index number of body j
XI-P-I	ξ -coordinate of point P on body i
ETA-P-I	η -coordinate of point P on body i
XI-P-J	ξ -coordinate of point P on body j
ETA-P-J	η -coordinate of point P on body j
K	Spring stiffness
D	Damping coefficient
Fa	Actuator force (- for push, + for pull)
L0	Undeformed length of the spring

(n) Rotational Spring-Damper

Refer to section 9.2.6 to 9.2.7 of the textbook (P.E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988), for detailed description of this force element.

The data lines for rotational spring-damper are repeated NRS times, and are as follows:

I J K D

where,

I	Index number of body i
J	Index number of body j
K	Spring stiffness
D	Damping coefficient

(o) Reaction External Forces

In this data line the program reads the index number of the reference bodies, which have reaction external acting on. For example, considering a biomechanical system of a human body, during the gait cycle there are ground reaction forces acting on foot segment, so this segment is a reference body.

The data lines for reference bodies are repeated NBEF times, and are as follows:

I

where,

I Index number of reference body i , which has reaction external forces acting on

The program also reads the time and reaction external forces data, which are introduced by user as discrete data and then interpolated by MUBODYNA. This data line includes NBEF \times 4+1 columns, which correspond to time and four parameters for each reference body, and NFD rows. The four parameters are, respectively, FX $_i$, FY $_i$, XI_P_I, ETA_P_I, and are defined as follow:

FX$_i$	X -reaction external force acting on the CM of body i
FY$_i$	Y -reaction external force acting on the CM of body i
XI_P_I	ζ -coordinate of the reaction external force application point P on body i
ETA_P_I	η -coordinate of the reaction external force application point P on body i

(p) Point of Interest

In this data line the program is instructed to output the coordinates, velocities, and accelerations of one or more points of interest attached to one or more rigid bodies. It should be noted that MUBODYNA automatically outputs the coordinates, velocities, and accelerations of the origins of the bodies.

The data lines for points of interest are repeated NP times, and are as follows:

I XI-P-I ETA-P-I

where,

I	Index number of body i
XI-P-I	ζ -coordinate of point P on body i
ETA-P-I	η -coordinate of point P on body i

(q) Time Parameters

It is important to select the time step adequately. A very small time step yields a more accurate result and, of course, it requires more computation time. A large time step is computationally more efficient but it may cause the numerical integration algorithm to yield erroneous results. The selection of the time step must be based on the highest frequency of the system response. However, since we deal with nonlinear differential equations, the system frequencies vary as the system orientation changes. Therefore, time step selection is not an easy task.

In rotating bodies, such as the slider crank mechanism, a reasonable choice for time step is a value that allows the crank not rotate more than 10 degrees in one time step.

In MUBODYNA the time parameters for dynamic analysis of a biomechanical system is given as follows:

TIME START TIME END TIME STEP INTEGRATION METHOD ALPHA BETA

where,

TIME START	Starting time
TIME END	Final time
TIME STEP	Time increment
INTEGRATION METHOD	A flag indicating what type of integrator method is used
ALPHA	α Baumgarte parameters
BETA	β Baumgarte parameters

Flag **INTEGRATION METHOD** controls what type of integration subroutine is used accordingly to:

EQ.1:	Fourth-order Runge-Kutta method
EQ.2:	Fourth-order Kutta-Merson method
EQ.3:	Fifth/sixth-order Runge-Kutta-Verner method
EQ.4:	Adams-Moulton's or Gear's BDF methods
EQ.5:	Adams predictor-corrector or Gear's methods
EQ.6:	Adams predictor-corrector formulas (ODE/DE)

6. OUTPUT FILE

The output results are written on a text file specified by the user. The results are reported with specified format and divided in two sections.

NUBODYNA report' first section compiles all the input data and is entitled "Multibody System Description".

The second section is entitled "Forward Dynamic Analysis" and reports the values of coordinates, velocities, and accelerations for all bodies in the system at the end of each time step. In addition, at this section, the global coordinates, velocities, and accelerations of the special points of interest are reported.

Moreover, if there is any translational spring-damper-actuator element in the system, the dynamic response of these elements, like the deformed length of the spring l , the time rate of change in length l' (dl/dt), the spring force f_s , and the damper force f_a , is also potted.

The total forces and torques on bodies' CM are calculated and reported, as well as the reaction external forces at the kinematic joints (including the driving constraints).

7. APPLICATION EXAMPLE 2: ½ HAT-LEG GUIDANCE MODEL

In this section, a biomechanical model of the human body is used in dynamic analysis corresponding to the stride period (Winter, 2005). The model is defined using 4 rigid bodies, corresponding to: **(1)** ½ Right HAT (acronym for Head-Arms-Trunk), **(2)** Right Thigh, **(3)** Right Leg and **(4)** Right Foot.

The bodies are connected by 3 ideal revolute joints, which correspond to hip, knee and ankle joints. Figure 7 illustrates schematically the configuration of the system, within the representation of a complete biomechanical model to perform gait analysis.

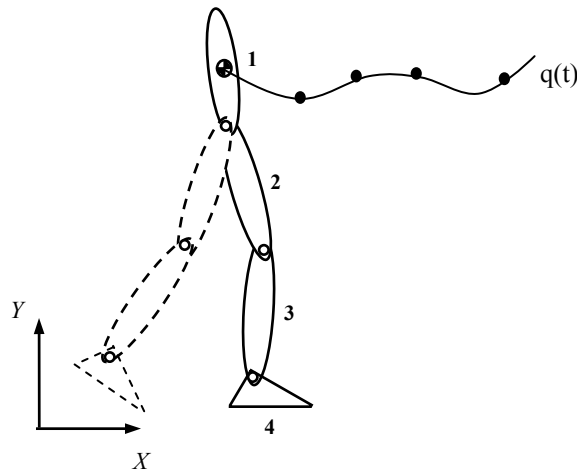


Figure 7: Configuration of the ½ HAT-Leg Guidance Model.

In general the number of guiding constraints must be equal to number of bodies plus 2 ($NGC = NB+2$). So, in this example are 6 guiding constraints: 2 guiding constraints for the CM (x,y) of the ½ HAT segment (main body) and 4 guiding constraints for ϕ of all bodies.

For this dynamic analysis, it is also considered the existence of reaction external forces acting on the right foot segment, which are the result of the interaction between the foot and the ground along stance period. The force value and application point vary with time.

For one complete gait cycle, 0.957 seconds of simulation time is required. In the dynamic simulation performed a time increment of 0.0145s is used, which represents the double of the input frequency data.

The fourth-order Runge-Kutta algorithm is employed to carry out the integration of the equations of motion. The explicit Runge-Kutta method is easy to implement, because it only require function evaluations, and is self-starting, meaning that it does not need any other algorithm or technique to start the integration process. However, the error control is a very difficult task. Stabilizing values α and β are equal to zero.

In short, the MUBODYNA input file that describes this biomechanical model is as follows:

```

GENERIC_INFORMATION - Number of elements of the Multibody System
  4 Rigid bodies
  3 Ideal revolute joints
  0 Clearance revolute joints
  0 Translational joints
  0 Ground constraints
  0 Simple constraints
  0 Driving constraints
  6 Guiding constraints
34 Guiding data for the guiding constraints
  0 Translational spring-damper-actuator elements
  0 Rotational spring-damper elements
  1 Bodies subjected to external applied forces
34 Force data of the external applied forces
  0 Points of interest

INITIAL_CONDITIONS - Global cartesian coordinates on the positions and velocities
  0.47266  1.07997  1.48527  1.38137  0.05727  0.73275

  0.43026  0.65208  1.44339  2.05470  -0.01989  3.39980

  0.27187  0.36231  0.69464  2.42049  0.18498  -2.07573

  0.08941  0.15333  1.49400  1.87062  0.57264  -3.98255

INERTIA_PROPERTIES - Mass and moment of inertia of the bodies
19.2213  1.03923
  5.6700  0.05836
  2.6365  0.04005
  0.8222  0.00273

CONSTANT_APPLIED_FORCES - Constant forces and moments acting on the bodies
  0.0000  0.0000  0.0000
  0.0000  0.0000  0.0000
  0.0000  0.0000  0.0000
  0.0000  0.0000  0.0000

IDEAL_REVOLUTE_JOINTS - Body numbers i and j and local coordinates of the points Pi and Pj
  1  2  -0.294065  0.000000  0.136005  0.000000
  2  3  -0.178095  0.000000  0.176707  0.000000
  3  4  -0.231393  0.000000  0.061050  0.000000

GUIDING_CONSTRAINTS - Guiding constraints
  1  1
  1  2
  1  3
  2  3
  3  3
  4  3
  0.0000  0.4726600  1.0799700  1.4852700  1.4433900  0.694640  1.49400
  0.0290  0.5123842  1.0821876  1.5044679  1.5463198  0.645770  1.40499
  0.0580  0.5506321  1.0884829  1.5152255  1.6637434  0.652280  1.43580
  0.0870  0.5868407  1.0976317  1.5240338  1.7733127  0.718770  1.56062
  0.1160  0.6212848  1.1077698  1.5368431  1.8717503  0.832720  1.74493
  0.1450  0.6556406  1.1162812  1.5529222  1.9448318  0.973740  1.94411
  0.1740  0.6894261  1.1222490  1.5760323  1.9921971  1.134060  2.14509
  0.2030  0.7236148  1.1243847  1.6011326  2.0191265  1.309260  2.34672
  0.2320  0.7596122  1.1224025  1.6215625  2.0221472  1.476670  2.53878
  0.2610  0.7978007  1.1163604  1.6371329  2.0075002  1.645450  2.73484
  0.2900  0.8383816  1.1081384  1.6469929  1.9803837  1.789340  2.90434
  0.3190  0.8808620  1.0999430  1.6560229  1.9527588  1.892510  3.02513
  0.3480  0.9245450  1.0936675  1.6694034  1.9307442  1.933960  3.06950
  0.3770  0.9686276  1.0901530  1.6850842  1.9176810  1.908400  3.02512
  0.4060  1.0147672  1.0899441  1.6962949  1.9047969  1.846320  2.91199
  0.4350  1.0613119  1.0939473  1.6976352  1.8935726  1.773470  2.77586
  0.4640  1.1074226  1.1010326  1.6884156  1.8869193  1.695610  2.67052
  0.4930  1.1530574  1.1086842  1.6710754  1.8729858  1.620690  2.62145
  0.5220  1.1968395  1.1147232  1.6545755  1.8369813  1.563330  2.61238
  0.5510  1.2384416  1.1196232  1.6428564  1.7786852  1.517370  2.61237
  0.5800  1.2774853  1.1230728  1.6288571  1.7162699  1.481200  2.60750
  0.6090  1.3166557  1.1256955  1.6053272  1.6577946  1.447950  2.59658
  0.6380  1.3565860  1.1264192  1.5747573  1.6037786  1.413850  2.57540
  0.6670  1.3966385  1.1238106  1.5409875  1.5562531  1.380390  2.54895
  0.6960  1.4377919  1.1189573  1.5065475  1.5087366  1.346480  2.52779
  0.7250  1.4791342  1.1135043  1.4753477  1.4593800  1.312460  2.51075
  0.7540  1.5212305  1.1079904  1.4473382  1.4075729  1.274960  2.48927
  0.7830  1.5648419  1.1025457  1.4195580  1.3574251  1.232080  2.45817
  0.8120  1.6082932  1.0974443  1.3966583  1.3179277  1.176170  2.40567
  0.8410  1.6522911  1.0924644  1.3805681  1.2907600  1.109620  2.31587
  0.8700  1.6968829  1.0878817  1.3773982  1.2766018  1.032720  2.18580
  0.8990  1.7436797  1.0833713  1.3890785  1.2806636  0.941990  2.01053
  0.9280  1.7911596  1.0795297  1.4136488  1.3078647  0.844440  1.79696
  0.9570  1.8355846  1.0776201  1.4475594  1.3618779  0.753260  1.58027

```

EXTERNAL_APPLIED_FORCES

```

4
0.000 0.0000 0.0000 0.0000 0.0000
0.029 0.0000 0.0000 0.0000 0.0000
0.058 0.0000 0.0000 0.0000 0.0000
0.087 0.0000 0.0000 0.0000 0.0000
0.116 0.0000 0.0000 0.0000 0.0000
0.145 0.0000 0.0000 0.0000 0.0000
0.174 0.0000 0.0000 0.0000 0.0000
0.203 0.0000 0.0000 0.0000 0.0000
0.232 0.0000 0.0000 0.0000 0.0000
0.261 0.0000 0.0000 0.0000 0.0000
0.290 0.0000 0.0000 0.0000 0.0000
0.319 0.0000 0.0000 0.0000 0.0000
0.348 0.0000 0.0000 0.0000 0.0000
0.377 0.0000 0.0000 0.0000 0.0000
0.406 -4.200 87.100 1.2440 0.0000
0.435 -74.00 192.60 1.2910 0.0000
0.464 -102.7 404.20 1.3010 0.0000
0.493 -114.2 521.70 1.3110 0.0000
0.522 -98.10 604.50 1.3160 0.0000
0.551 -62.00 558.10 1.3220 0.0000
0.580 -39.80 473.40 1.3330 0.0000
0.609 -27.00 400.30 1.3450 0.0000
0.638 -18.10 364.00 1.3690 0.0000
0.667 -16.80 366.50 1.3840 0.0000
0.696 -14.50 386.10 1.3960 0.0000
0.725 -3.800 418.80 1.4170 0.0000
0.754 8.7000 465.80 1.4240 0.0000
0.783 26.800 523.90 1.4320 0.0000
0.812 54.800 576.80 1.4400 0.0000
0.841 79.600 608.60 1.4470 0.0000
0.870 101.50 602.30 1.4550 0.0000
0.899 115.60 530.30 1.4630 0.0000
0.928 105.20 377.30 1.4700 0.0000
0.957 65.700 190.10 1.4780 0.0000

```

TIME_PARAMETERS

```

0.0000 0.9570 0.00145 1 0 0

```

The output results for the two first time and the two final steps are presented below.

*** MULTIBODY SYSTEM DESCRIPTION ***

GENERIC_INFORMATION

```

4 Rigid bodies
3 Ideal revolute joints
0 Clearance revolute joint
0 Translational joints
0 Ground constraints
0 Simple constraints
0 Driving constraints
6 Guiding constraints
34 Guiding data for the guiding constraints
0 Translational spring-damper-actuators elements
0 Rotational spring-damper elements
1 Bodies subjected to external applied forces
34 Force data of the external applied forces
0 Points of interest

```

INITIAL_CONDITIONS

```

X Y PHI XD YD PHID
0.4727 1.0800 1.4853 1.3814 0.0573 0.7328
0.4303 0.6521 1.4434 2.0547 -0.0199 3.3998
0.2719 0.3623 0.6946 2.4205 0.1850 -2.0757
0.0894 0.1533 1.4940 1.8706 0.5726 -3.9825

```

INERTIA_PROPERTIES

```

m I
19.2213 1.0392
5.6700 0.0584
2.6365 0.0401
0.8222 0.0027

```

CONSTANT_APPLIED_FORCES

```

FX FY M
0.0000 -188.5610 0.0000
0.0000 -55.6227 0.0000
0.0000 -25.8641 0.0000
0.0000 -8.0658 0.0000

```

IDEAL_REVOLUTE_JOINTS

i	j	XI-P-I	ETA-P-I	XI-P-J	ETA-P-J
1	2	-0.2941	0.0000	0.1360	0.0000
2	3	-0.1781	0.0000	0.1767	0.0000
3	4	-0.2314	0.0000	0.0611	0.0000

GUIDING_CONSTRAINTS

i Direction (1=X, 2=Y, 3=PHI)

1	1
1	2
1	3
2	3
3	3
4	3

Time	Guid_1	Guid_2	Guid_3	...		
0.0000	0.4727	1.0800	1.4853	1.4434	0.6946	1.4940
0.0290	0.5124	1.0822	1.5045	1.5463	0.6458	1.4050
0.0580	0.5506	1.0885	1.5152	1.6637	0.6523	1.4358
0.0870	0.5868	1.0976	1.5240	1.7733	0.7188	1.5606
0.1160	0.6213	1.1078	1.5368	1.8718	0.8327	1.7449
0.1450	0.6556	1.1163	1.5529	1.9448	0.9737	1.9441
0.1740	0.6894	1.1222	1.5760	1.9922	1.1341	2.1451
0.2030	0.7236	1.1244	1.6011	2.0191	1.3093	2.3467
0.2320	0.7596	1.1224	1.6216	2.0221	1.4767	2.5388
0.2610	0.7978	1.1164	1.6371	2.0075	1.6455	2.7348
0.2900	0.8384	1.1081	1.6470	1.9804	1.7893	2.9043
0.3190	0.8809	1.0999	1.6560	1.9528	1.8925	3.0251
0.3480	0.9245	1.0937	1.6694	1.9307	1.9340	3.0695
0.3770	0.9686	1.0902	1.6851	1.9177	1.9084	3.0251
0.4060	1.0148	1.0899	1.6963	1.9048	1.8463	2.9120
0.4350	1.0613	1.0939	1.6976	1.8936	1.7735	2.7759
0.4640	1.1074	1.1010	1.6884	1.8869	1.6956	2.6705
0.4930	1.1531	1.1087	1.6711	1.8730	1.6207	2.6214
0.5220	1.1968	1.1147	1.6546	1.8370	1.5633	2.6124
0.5510	1.2384	1.1196	1.6429	1.7787	1.5174	2.6124
0.5800	1.2775	1.1231	1.6289	1.7163	1.4812	2.6075
0.6090	1.3167	1.1257	1.6053	1.6578	1.4480	2.5966
0.6380	1.3566	1.1264	1.5748	1.6038	1.4139	2.5754
0.6670	1.3966	1.1238	1.5410	1.5563	1.3804	2.5490
0.6960	1.4378	1.1190	1.5065	1.5087	1.3465	2.5278
0.7250	1.4791	1.1135	1.4753	1.4594	1.3125	2.5107
0.7540	1.5212	1.1080	1.4473	1.4076	1.2750	2.4893
0.7830	1.5648	1.1025	1.4196	1.3574	1.2321	2.4582
0.8120	1.6083	1.0974	1.3967	1.3179	1.1762	2.4057
0.8410	1.6523	1.0925	1.3806	1.2908	1.1096	2.3159
0.8700	1.6969	1.0879	1.3774	1.2766	1.0327	2.1858
0.8990	1.7437	1.0834	1.3891	1.2807	0.9420	2.0105
0.9280	1.7912	1.0795	1.4136	1.3079	0.8444	1.7970
0.9570	1.8356	1.0776	1.4476	1.3619	0.7533	1.5803

APPLIED_FORCES

i

4

Time	FX_1	FY_1	X_P_1	Y_P_1	...
0.0000	0.0000	0.0000	0.0000	0.0000	
0.0290	0.0000	0.0000	0.0000	0.0000	
0.0580	0.0000	0.0000	0.0000	0.0000	
0.0870	0.0000	0.0000	0.0000	0.0000	
0.1160	0.0000	0.0000	0.0000	0.0000	
0.1450	0.0000	0.0000	0.0000	0.0000	
0.1740	0.0000	0.0000	0.0000	0.0000	
0.2030	0.0000	0.0000	0.0000	0.0000	
0.2320	0.0000	0.0000	0.0000	0.0000	
0.2610	0.0000	0.0000	0.0000	0.0000	
0.2900	0.0000	0.0000	0.0000	0.0000	
0.3190	0.0000	0.0000	0.0000	0.0000	
0.3480	0.0000	0.0000	0.0000	0.0000	
0.3770	0.0000	0.0000	0.0000	0.0000	
0.4060	-4.2000	87.1000	1.2440	0.0000	
0.4350	-74.0000	192.6000	1.2910	0.0000	
0.4640	-102.7000	404.2000	1.3010	0.0000	
0.4930	-114.2000	521.7000	1.3110	0.0000	
0.5220	-98.1000	604.5000	1.3160	0.0000	
0.5510	-62.0000	558.1000	1.3220	0.0000	
0.5800	-39.8000	473.4000	1.3330	0.0000	
0.6090	-27.0000	400.3000	1.3450	0.0000	
0.6380	-18.1000	364.0000	1.3690	0.0000	
0.6670	-16.8000	366.5000	1.3840	0.0000	
0.6960	-14.5000	386.1000	1.3960	0.0000	
0.7250	-3.8000	418.8000	1.4170	0.0000	
0.7540	8.7000	465.8000	1.4240	0.0000	
0.7830	26.8000	523.9000	1.4320	0.0000	
0.8120	54.8000	576.8000	1.4400	0.0000	
0.8410	79.6000	608.6000	1.4470	0.0000	
0.8700	101.5000	602.3000	1.4550	0.0000	
0.8990	115.6000	530.3000	1.4630	0.0000	
0.9280	105.2000	377.3000	1.4700	0.0000	
0.9570	65.7000	190.1000	1.4780	0.0000	

TIME_PARAMETERS

TIME_START	TIME_END	TIME_STEP	INTEGRATION_METHOD	ALPHA	BETA
0.0000	0.9570	0.0014	1	5	5

*** FORWARD DYNAMIC ANALYSIS ***

TIME = 0.00000

BODY	X	Y	PHI	XD	YD	PHID	XDD	YDD	PHIDD
1	0.473	1.080	1.485	1.381	0.057	0.733	-0.018	-0.123	0.000
2	0.430	0.652	1.443	2.055	-0.020	3.400	0.195	1.594	0.001
3	0.272	0.362	0.695	2.420	0.185	-2.076	1.042	4.123	0.000
4	0.089	0.153	1.494	1.871	0.573	-3.983	1.884	5.727	0.020

REACTION FORCES

JOINT NO.	i	FX-i	FY-i	M-i	j	FX-j	FY-j	M-j
REV. 1	1	-5.404	-114.168	1.285	2	5.404	114.168	1.244
REV. 2	2	-4.296	-49.509	0.361	3	4.296	49.509	6.235
REV. 3	3	-1.549	-12.774	2.042	4	1.549	12.774	-0.034

CONSTRAINT VIOLATIONS

CONSTRAINT	Position violations	Velocity violations
1	-0.7058E-06	0.5628E-06
2	-0.2800E-05	0.1441E-05
3	-0.2875E-06	0.3795E-06
4	0.6775E-05	-0.8061E-06
5	0.6076E-06	0.1619E-05
6	-0.6780E-05	0.7615E-05
7	0.0000E+00	0.1796E-02
8	0.0000E+00	0.1228E-01
9	0.0000E+00	-0.3617E-04
10	0.0000E+00	-0.5083E-04
11	0.0000E+00	0.6715E-05
12	0.0000E+00	-0.2035E-02

TIME = 0.47995

BODY	X	Y	PHI	XD	YD	PHID	XDD	YDD	PHIDD
1	1.133	1.106	1.679	1.575	0.265	-0.622	-1.513	-1.243	-4.083
2	1.206	0.684	1.881	1.330	0.225	-0.483	-5.287	-2.051	-19.758
3	1.275	0.338	1.653	0.791	0.161	-2.597	-5.956	-1.669	15.851
4	1.347	0.078	2.637	0.146	0.028	-1.569	-0.716	3.598	62.479

REACTION FORCES

JOINT NO.	i	FX-i	FY-i	M-i	j	FX-j	FY-j	M-j
REV. 1	1	-64.648	401.668	-6.102	2	64.648	-401.668	8.316
REV. 2	2	-94.625	445.662	8.199	3	94.625	-445.662	-10.210
REV. 3	3	-110.329	467.127	-16.584	4	110.329	-467.127	21.705

CONSTRAINT VIOLATIONS

CONSTRAINT	Position violations	Velocity violations
1	-0.1932E-06	0.6971E-06
2	-0.8011E-06	0.2865E-05
3	-0.7226E-07	0.2657E-06
4	0.2054E-05	-0.7274E-05
5	0.2579E-06	-0.8647E-06
6	-0.1761E-05	0.6416E-05
7	0.7821E-04	-0.2281E-03
8	0.5346E-03	-0.1559E-02
9	-0.1575E-05	0.4594E-05
10	-0.2214E-05	0.6456E-05
11	0.2924E-06	-0.8523E-06
12	-0.8863E-04	0.2585E-03

TIME = 0.95700

BODY	X	Y	PHI	XD	YD	PHID	XDD	YDD	PHIDD
1	1.836	1.078	1.448	1.503	-0.050	1.232	0.000	0.001	0.000
2	1.771	0.653	1.362	2.137	-0.153	2.063	0.175	1.010	0.000
3	1.605	0.358	0.753	2.125	0.168	-3.076	1.552	2.896	0.000
4	1.437	0.138	1.580	1.184	0.682	-7.430	3.117	7.764	0.000

REACTION FORCES

JOINT NO.	i	FX-i	FY-i	M-i	j	FX-j	FY-j	M-j
REV. 1	1	58.052	80.800	14.021	2	-58.052	-80.800	5.445
REV. 2	2	59.045	142.152	5.036	3	-59.045	-142.152	-11.187
REV. 3	3	63.137	175.651	-19.655	4	-63.137	-175.651	3.956

CONSTRAINT VIOLATIONS

CONSTRAINT	Position violations	Velocity violations
1	-0.2960E-07	0.1230E-06
2	-0.1239E-06	0.5134E-06
3	-0.1086E-07	0.4491E-07
4	0.3206E-06	-0.1329E-05
5	0.4241E-07	-0.1729E-06
6	-0.2673E-06	0.1115E-05
7	0.1436E-04	-0.5679E-04
8	0.9814E-04	-0.3881E-03
9	-0.2892E-06	0.1144E-05
10	-0.4063E-06	0.1607E-05
11	0.5367E-07	-0.2123E-06
12	-0.1627E-04	0.6435E-04

8. REFERENCES

- BAUMGARTE, J., 1972, "Stabilization of Constraints and Integrals of Motion in Dynamical Systems", *Computer Methods in Applied Mechanics and Engineering*, 1, pp. 1-16.
- BAYO, E., GARCIA de JALON, J., and SERNA, A., 1988, "A Modified Lagrangian Formulation for the Dynamic Analysis of Constrained Mechanical Systems", *Computer Methods in Applied Mechanics and Engineering*, 71, pp. 183-195.
- CHANG, C.O., and NIKRAVESH, P.E., 1985, "An Adaptive Constraint Violation Stabilization Method for Dynamic Analysis of Mechanical Systems", *Journal of Mechanisms, Transmissions, and Automation in Design*, 107, pp. 488-492.
- COCHIN, I., and CADWALLENDER, W., 1997, "Analysis and Design of Dynamic Systems", 3rd Edition, Addison-Wesley, New Jersey.
- FLORES, P., AMBRÓSIO, J., CLARO, J.C.P., LANKARANI, H.M., 2008, "Kinematics and Dynamics of Multibody Systems with Imperfect Joints – Models and Case Studies", *Lecture Notes in Applied and Computational Mechanics*, vol. 34, Springer-Verlag, Berlin Heidelberg.
- GARCIA de JALON, J., and BAYO, E., 1994, "Kinematic and Dynamic Simulations of Multibody Systems", Springer-Verlag, New York.
- GEAR, C.W., 1981, "Numerical Solution of Differential-Algebraic Equations", *IEEE Transactions on Circuit Theory*, Vol. CT-18, pp. 89-95.
- NIKRAVESH P.E., 1988, "Computer-Aided Analysis of Mechanical Systems", Prentice Hall, Englewood Cliffs, New Jersey.
- SHAMPINE, L., and GORDON, M., 1975, "Computer Solution of Ordinary Differential Equations: The Initial Value Problem", Freeman, San Francisco, California.
- WEHAGE, R.A., and HAUG, E.J., 1982, "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Systems", *Journal of Mechanical Design*, 104, pp. 247-255.
- WINTER, D.A., 2005, "Biomechanics and Motor Control of Human Movement", 3rd Edition, John Wiley & Sons, Inc.

9. FLOWCHART OF THE MUBODYNA PROGRAM

```
START
  USE MUBODYNA_ARRAYS
  CALL MUBODYNA_UNITS
  CALL GENERAL_INFORMATION
  CALL MUBODYNA_ALLOCATE
  CALL SPECIFIC_INFORMATION
    CALL INPUT_INITIAL_CONDITIONS
    CALL INPUT_INERTIA_PROPERTIES
    CALL INPUT_CONSTANT_APPLIED_FORCES
    CALL INPUT_IDEAL_REVOLUTE_JOINTS
    CALL INPUT_CLEARANCE_REVOLUTE_JOINTS
    CALL INPUT_MECHANICAL_PROPERTIES
    CALL INPUT_TRANSLATIONAL_JOINTS
    CALL INPUT_GROUND_CONSTRAINTS
    CALL INPUT_SIMPLE_CONSTRAINTS
    CALL INPUT_DRIVING_CONSTRAINTS
    CALL INPUT_GUIDING_CONSTRAINTS
    CALL INPUT_TRANSLATIONAL_SPRINGS
    CALL INPUT_ROTATIONAL_SPRINGS
    CALL INPUT_EXTERNAL_APPLIED_FORCES
    CALL INPUT_POINTS_OF_INTEREST
    CALL INPUT_TIME_PARAMETERS
  CALL DYNAMIC_ANALYSIS
    CALL TRANSFER_POSITIONS_VELOCITIES
    INTEGRATION_METHOD=1  CALL INTEGRATOR_RUNGE_KUTTA
    INTEGRATION_METHOD=2  CALL INTEGRATOR_KUTTA_MERSON
    INTEGRATION_METHOD=3  CALL INTEGRATOR_RUNGE_KUTTA_VERNER
    INTEGRATION_METHOD=4  CALL INTEGRATOR_DIVPAG
    INTEGRATION_METHOD=5  CALL INTEGRATOR_DGEAR
    INTEGRATION_METHOD=6  CALL INTEGRATOR_ODE_PREDICTOR_CORRECTOR
    CALL EQUATIONS_OF_MOTION
      CALL TRANSFER_Y
      CALL SOLVE_EQUATIONS_OF_MOTION
        CALL TRIGONOMETRIC_FUNCTIONS
        CALL EVALUATE_FUNCTIONS
          CALL FUNCTIONS_REVOLUTE_JOINTS
          CALL FUNCTIONS_TRANSLATIONAL_JOINTS
          CALL FUNCTIONS_GROUND_CONSTRAINTS
          CALL FUNCTIONS_SIMPLE_CONSTRAINTS
          CALL FUNCTIONS_DRIVING_CONSTRAINTS
          CALL FUNCTIONS_GUIDING_CONSTRAINTS
        CALL CONSTRAINT_VIOLATIONS
        CALL BODIES_APPLIED_FORCES
          CALL BODIES_CONSTANT_FORCES
          CALL TRANSLATIONAL_SPRING_FORCES
          CALL ROTATIONAL_SPRING_MOMENTS
          CALL EXTERNAL_REACTION_FORCES
          CALL REVOLUTE_CLEARANCE_JOINTS_FORCES
          FORCE_MODEL=0  CALL LUBRICATION_FORCES
          FORCE_MODEL=1  CALL DRY_CONTACT_FORCES
          CALL ECCENTRICITY
          CALL PENETRATION
          DELTA>0
            CALL NORMAL_VECTOR
            CALL TANGENTIAL_VECTOR
            CALL CONTACT_POINTS
            CALL NORMAL_FORCE
              FORCE_MODEL=1  CALL HERTZ
              FORCE_MODEL=2  CALL LANKARANI_NIKRAVESH
              FORCE_MODEL=3  CALL HUNT_CROSSLEY
            CALL TANGENTIAL_FORCE
          CALL JOIN_G_FORCES_VECTOR_GAMMA_VECTOR
        CALL COEFFICIENTS_MATRIX_A
        CALL SOLVE_LINEAR_EQUATIONS_SYSTEM
          CALL LU_FACTORIZATION
        CALL TRANSFER_ACCELERATIONS_LAGRANGE_MULTIPLIERS
      CALL TRANSFER_VELOCITIES_ACCELERATIONS
    CALL REPORT_RESULTS
      CALL REPORT_KINEMATIC_OF_BODIES_DATA
      CALL REPORT_POINTS_OF_INTEREST_DATA
      CALL REPORT_TRANSLATIONAL_SPRINGS_DATA
      CALL REPORT_REACTION_FORCES_DATA
      CALL REPORT_CONSTRAINT_VIOLATIONS_DATA
    CALL MUBODYNA_DEALLOCATE
STOP
```