# Simulation of a double pendulum

25th October 2003

## 1 General data of the studied mechanism

The system comprises 2 bodies (defined by the global variable `nbrbody`). Each body is called $S_j$ ($j$ from 0 to 1). The number of degrees of freedom of the system is 2 (`nbrdof`). The configuration parameters are denoted by $q_i$ ($i$ from 0 to 1).

The inertial data, given by the user, consist of the mass $m_{Si}$ and the inertia tensor $\Phi_{G,Si}$ of each body $i$ expressed with respect to the center of gravity.

$$m_{S0} = 1.1\,kg$$
$$m_{S1} = 0.9\,kg$$

$$\Phi_{G,S0} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.132 \end{pmatrix} \quad \text{, in } kg.m^2$$

$$\Phi_{G,S1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.09075 \end{pmatrix} \quad \text{, in } kg.m^2$$

## 2 Complete kinematics calculed by CAGeM

The following parameters have been calculated from the user's file `dp3.mu` and with a *CPU* time of 1 second.

### Homogeneous transformation matrix of each body

$$T_{0G,S0} = \begin{pmatrix} \cos(q_0) & -\sin(q_0) & 0 & 0.6\sin(q_0) \\ \sin(q_0) & \cos(q_0) & 0 & -0.6\cos(q_0) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{0G,S1} = \begin{pmatrix} \cos(q_0+q_1) & -\sin(q_0+q_1) & 0 & 1.2\sin(q_0)+0.55\sin(q_0+q_1) \\ \sin(q_0+q_1) & \cos(q_0+q_1) & 0 & -1.2\cos(q_0)-0.55\cos(q_0+q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

**The time derivative of these matrices**

$$\dot{T}_{0G,S0} = \begin{pmatrix} -\dot{q}_0\sin(q_0) & -\dot{q}_0\cos(q_0) & 0 & 0.6\,\dot{q}_0\cos(q_0) \\ \dot{q}_0\cos(q_0) & -\dot{q}_0\sin(q_0) & 0 & 0.6\,\dot{q}_0\sin(q_0) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\dot{T}_{0G,S1} = \begin{pmatrix} -\dot{q}_0\sin(q_0+q_1)-\dot{q}_1\sin(q_0+q_1) & -\dot{q}_0\cos(q_0+q_1)-\dot{q}_1\cos(q_0+q_1) & 0 & 1.2\,\dot{q}_0\cos(q_0)+0.55\,\dot{q} \\ \dot{q}_0\cos(q_0+q_1)+\dot{q}_1\cos(q_0+q_1) & -\dot{q}_0\sin(q_0+q_1)-\dot{q}_1\sin(q_0+q_1) & 0 & 1.2\,\dot{q}_0\sin(q_0)+0.55\,\dot{q} \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \end{pmatrix}$$

**The absolute velocity of the center of gravity of each body**

$$\vec{v}_{G,S0} = \begin{pmatrix} 0.6\,\dot{q}_0\cos(q_0) \\ 0.6\,\dot{q}_0\sin(q_0) \\ 0 \end{pmatrix}$$

$$\vec{v}_{G,S1} = \begin{pmatrix} 1.2\,\dot{q}_0\cos(q_0)+0.55\,\dot{q}_0\cos(q_0+q_1)+0.55\,\dot{q}_1\cos(q_0+q_1) \\ 1.2\,\dot{q}_0\sin(q_0)+0.55\,\dot{q}_0\sin(q_0+q_1)+0.55\,\dot{q}_1\sin(q_0+q_1) \\ 0 \end{pmatrix}$$

**The absolute aceleration of the center of gravity of each body**

$$\vec{a}_{G,S0} = \begin{pmatrix} 0.6\,\ddot{q}_0\cos(q_0)-0.6\,\dot{q}_0^2\sin(q_0) \\ 0.6\,\ddot{q}_0\sin(q_0)+0.6\,\dot{q}_0^2\cos(q_0) \\ 0 \end{pmatrix}$$

$$\vec{a}_{G,S1} = \begin{pmatrix} 1.2\,\ddot{q}_0\cos(q_0)-1.2\,\dot{q}_0^2\sin(q_0)+0.55\,\ddot{q}_0\cos(q_0+q_1)+0.55\,\ddot{q}_1\cos(q_0+q_1)-0.55\,\dot{q}_0^2\sin(q_0+q_1)- \\ 1.2\,\ddot{q}_0\sin(q_0)+1.2\,\dot{q}_0^2\cos(q_0)+0.55\,\ddot{q}_0\sin(q_0+q_1)+0.55\,\ddot{q}_1\sin(q_0+q_1)+0.55\,\dot{q}_0^2\cos(q_0+q_1)+ \\ 0 \end{pmatrix}$$

**The rotation velocity of each body**

$$\vec{\omega}_{S0} = \begin{pmatrix} 0 \\ 0 \\ \dot{q}_0 \end{pmatrix}$$

$$\vec{\omega}_{S1} = \begin{pmatrix} 0 \\ 0 \\ \dot{q}_0 + \dot{q}_1 \end{pmatrix}$$

**The rotation acceleration of each body**

$$\vec{\dot{\omega}}_{S0} = \begin{pmatrix} 0 \\ 0 \\ \ddot{q}_0 \end{pmatrix}$$

$$\vec{\dot{\omega}}_{S1} = \begin{pmatrix} 0 \\ 0 \\ \ddot{q}_0 + \ddot{q}_1 \end{pmatrix}$$

# 3 Simulation

The routine `NewmarkIntegration` performs the integration of the equations of motion up to time *FinalTime* by regular time intervals equal to *StepSave* and with the maximum allowed time step *StepMax* defined in the file `dp3.cpp`. The following values are used:

- *FinalTime* equal to TimeFinal $s$,
- *StepSave* to 0.01 $s$,
- *StepMax* to 0.005 $s$.

The initial conditions are $q_1 = 1$, the others being equal to zero.

# 4 Results

The time evolution of the different configuration parameters and their first and second derivatives can be easily plotted by Gnuplot as you can see in figures 1 to 3 etablished by the following script:

```
reset
set xlabel "Time [s]"
set grid

set term postscript eps color "Times-Roman" 20
```

```
set output "figure1.eps"
set ylabel "displacements"
plot 'dp3.res' using 1:2 title 'q_0' with line , 'dp3.res' using 1:5 title 'q_1' with line
set term X11
replot
pause -1 'Next plot (velocity level)?'

set term postscript eps color "Times-Roman" 20
set output "figure2.eps"
set ylabel "velocities"
plot 'dp3.res' using 1:3 title 'qd_0' with line , 'dp3.res' using 1:6 title 'qd_1' with line
set term X11
replot
pause -1 'Next plot (acceleration level)?'

set term postscript eps color "Times-Roman" 20
set output "figure3.eps"
set ylabel "accelerations"
plot 'dp3.res' using 1:4 title 'qdd_0' with line , 'dp3.res' using 1:7 title 'qdd_1' with line
set term X11
replot
```

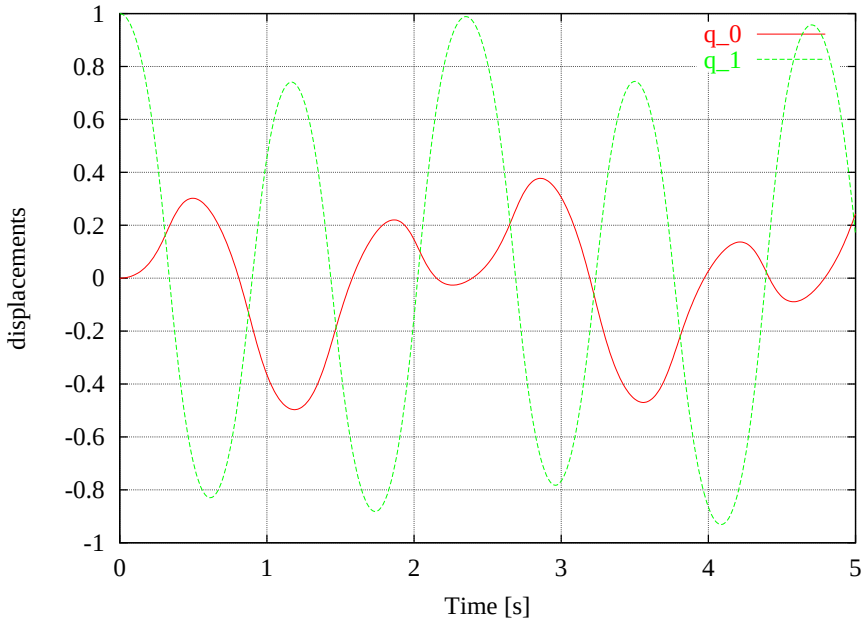


Figure 1: Time evolution of parameters

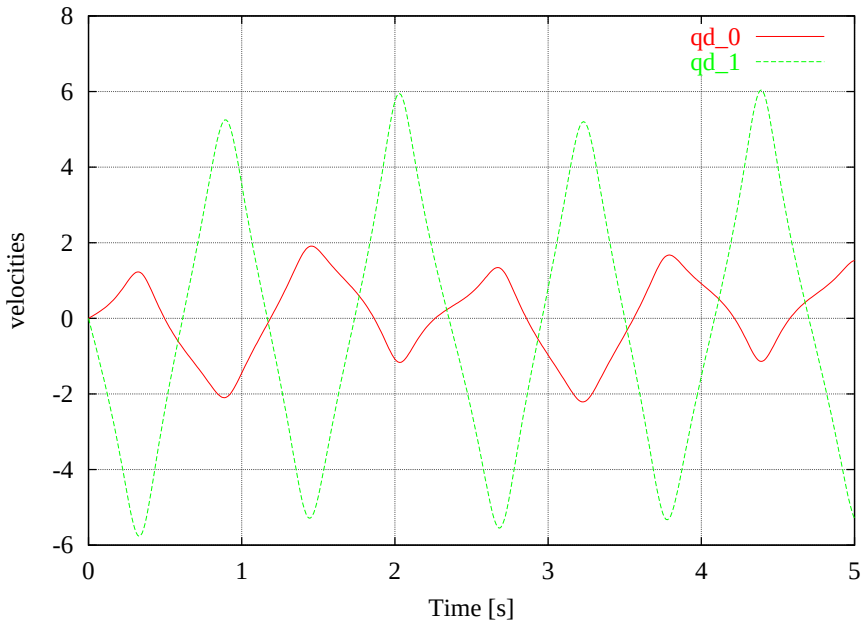


Figure 2: Time evolution of time derivatives of parameters

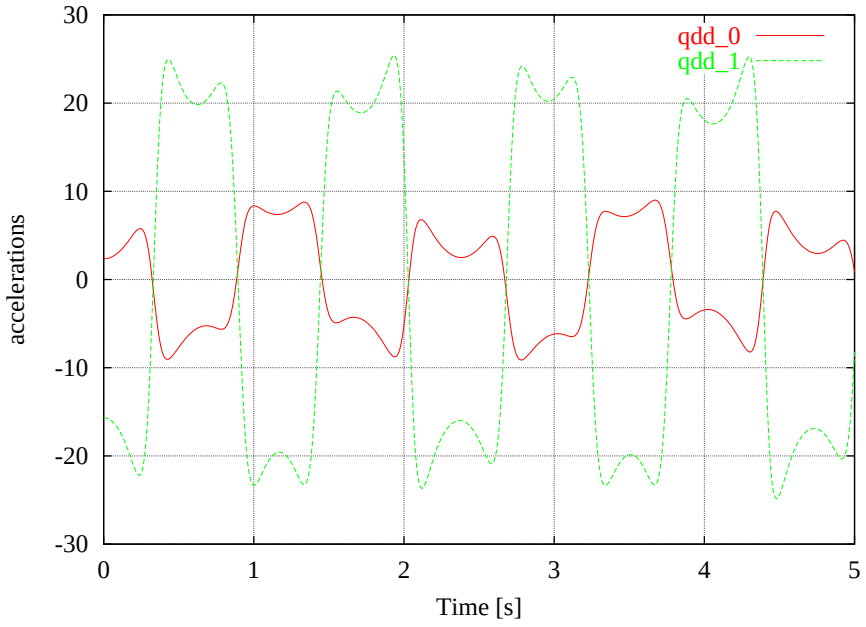


Figure 3: Time evolution of second time derivatives of parameters

# A User's MuPAD code

```
//

// Copyright (C) 2003 Olivier VERLINDEN
//     Service de Mecanique rationnelle, Dynamique et Vibrations
//     Faculte Polytechnique de Mons
//     31, Bd Dolez, 7000 MONS (Belgium)
//     Olivier.Verlinden@fpms.ac.be

// This file is part of EasyDyn

// EasyDyn is free software; you can redistribute it and/or modify it
// under the terms of the GNU General Public License as published by the
// Free Software Foundation; either version 2, or (at your option) any
// later version.

// EasyDyn is distributed in the hope that it will be useful, but WITHOUT
// ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
// FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
// for more details.

// You should have received a copy of the GNU General Public License
// along with EasyDyn; see the file COPYING.  If not, write to the Free
// Software Foundation, 59 Temple Place - Suite 330, Boston, MA  02111-1307, USA.
//
//                              FACULTE POLYTECHNIQUE DE MONS
//
//               service de Mécanique Rationnelle, Dynamique et Vibrations
//
//
//                    ------------------------------------------------
//                                fichier utilisateur
//                              (écrit en language MuPAD)
//
//
//                  Ir. Georges KOUROUSSIS - mars 2003
//
//
//

// Title of the application
title:="Simulation of a double pendulum":

// Definition of nbrdof  : Number of degrees of freedom
//               nbrbody : Number of bodies
//               nbrcont : Number of constraints (unused in this version).
nbrdof:= 2:
nbrbody:= 2:

// Gravity vector
gravity[1]:=0:
gravity[2]:=-9.81:
gravity[3]:=0:

// Eventual constants
l0:=1.2:
l1:=1.1:

// Inertia characteristics
mass[0]:=1.1:
mass[1]:=0.9:
```

```
Ixx[0]:=1:
Ixx[1]:=1:
Iyy[0]:=1:
Iyy[1]:=1:
Izz[0]:=l0^2/12*mass[0]:
Izz[1]:=l1^2/12*mass[1]:

// Definition of the position matrices
T0G[0] := Trotz(q[0]) * Tdisp(0,-l0/2,0):
T0G[1] := Trotz(q[0]) * Tdisp(0,-l0,0) * Trotz(q[1]) * Tdisp(0,-l1/2,0):

// Initial conditions
qi[1]:=1:

// Simulation parameters
FinalTime:=5:
StepSave:=0.01:
StepMax:=0.005:

SIMPLIFY:=1:
// Set FORCES to 1 in case you want to include dp3.AppEff.cpp into procedure
// AddAppliedEfforts() to define forces other than gravity
FORCES:=0:
// Set ANIM to 1 in case you want CaGEM to generate the skeleton code
// for visualization and animation of the system
ANIM:=1:
// Set STATIC to 1 in case you want CaGEM to generate the code
// to search for static equilibrium before integration
STATIC:=0:
// Set PLOT to 1 in case you want CaGEM to generate the GNUPLOT code
// to plot the evolution of position, velocity and acceeration
PLOT:=1:
// SET LATEX_FR to 1 if you want the LaTeX report in French
LATEX_FR:=1:
// SET LATEX_EN to 1 if you want the LaTeX report in English
LATEX_EN:=1:
```