

Simulation of a double pendulum

26 avril 2003

1 Données générales du mécanisme étudié

Le nombre de solides du système s'élève à 2 (défini par la variable globale `nbrbody+1`). Par la suite, ils se définissent par S_j avec j allant de 0 à 1. Le nombre de degrés de liberté s'élève, quant à lui, à 2 (`nbrdof+1`). Les paramètres de configuration sont donc définis par q_i (i allant de 0 à 1).

Les données inertielles, définies dans le fichier utilisateur, se résument aux masses m_{S_i} et aux tenseurs d'inertie Φ_{G,S_i} définis au centre de gravité de chaque solide concerné.

$$m_{S0} = 1.1 \text{ kg}$$

$$m_{S1} = 0.9 \text{ kg}$$

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

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

2 Paramètres cinématiques calculés sous MuPAD

Ces paramètres ont été calculés à partir du fichier utilisateur `dp3.mu` et ce, avec un temps de calcul *CPU* de 9 secondes.

Les matrices de transformation homogène de chaque solide

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

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

Leur dérivée par rapport au temps

$$T'_{0G,S0} = \begin{pmatrix} -\dot{q}_0 \sin(q_0) & -\dot{q}_0 \cos(q_0) & 0 & 0.6 \dot{q}_0 \cos(q_0) \\ \dot{q}_0 \cos(q_0) & -\dot{q}_0 \sin(q_0) & 0 & 0.6 \dot{q}_0 \sin(q_0) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$T'_{0G,S1} = \begin{pmatrix} -\dot{q}_0 \sin(q_0 + q_1) - \dot{q}_1 \sin(q_0 + q_1) & -\dot{q}_0 \cos(q_0 + q_1) - \dot{q}_1 \cos(q_0 + q_1) & 0 & 1.2 \dot{q}_0 \cos(q_0) + 0.55 \dot{q}_1 \cos(q_0 + q_1) \\ \dot{q}_0 \cos(q_0 + q_1) + \dot{q}_1 \cos(q_0 + q_1) & -\dot{q}_0 \sin(q_0 + q_1) - \dot{q}_1 \sin(q_0 + q_1) & 0 & 1.2 \dot{q}_0 \sin(q_0) + 0.55 \dot{q}_1 \sin(q_0 + q_1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Les vitesses des centres de gravité des solides

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

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

Les accélérations des centres de gravité des solides

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

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

Les vitesses de rotation des solides

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

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

Les accélérations de rotation des solides

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

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

3 Simulation du mécanisme

La simulation temporelle s'est faite via la procédure `NewmarkIntegration`(*TempsFinal* , *StepSave* , *StepMax*) du fichier `dp3.cpp` où :

- *TempsFinal* est la durée de la simulation (=5 s),
- *StepSave* est le pas de temps dans l'intégration numérique (=0.01 s),
- *StepMax* est un pas de temps limite (=0.005 s),

les conditions initiales étant $q_1 = 1$, les autres étant nulles.

4 Résultats

L'affichage des évolutions temporelles des différents paramètres de configuration ainsi que leurs dérivées première et seconde s'établit assez facilement sous `MatLab` comme nous le montrent les figures 1 à 3 créées par le code suivant :

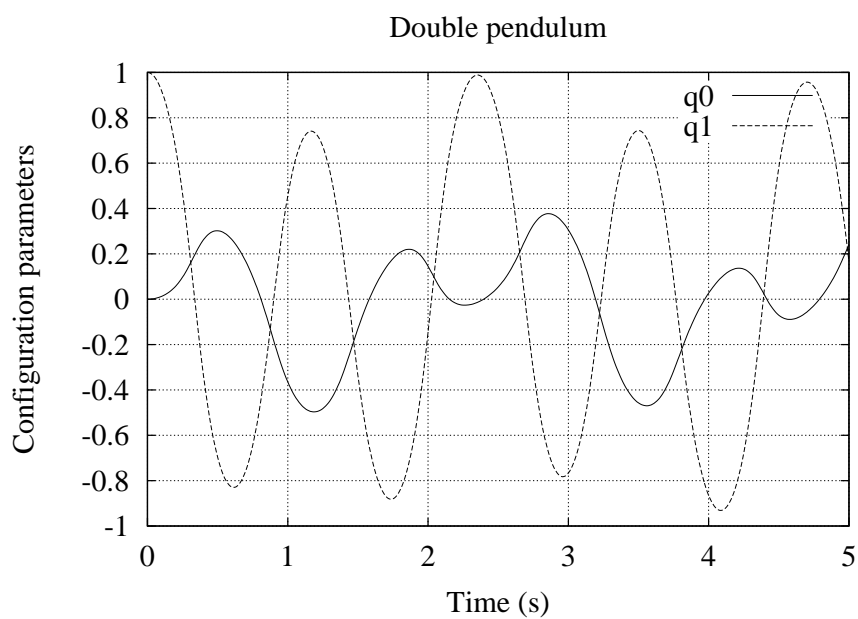


FIG. 1 – Evolution temporelle des paramètres de configuration

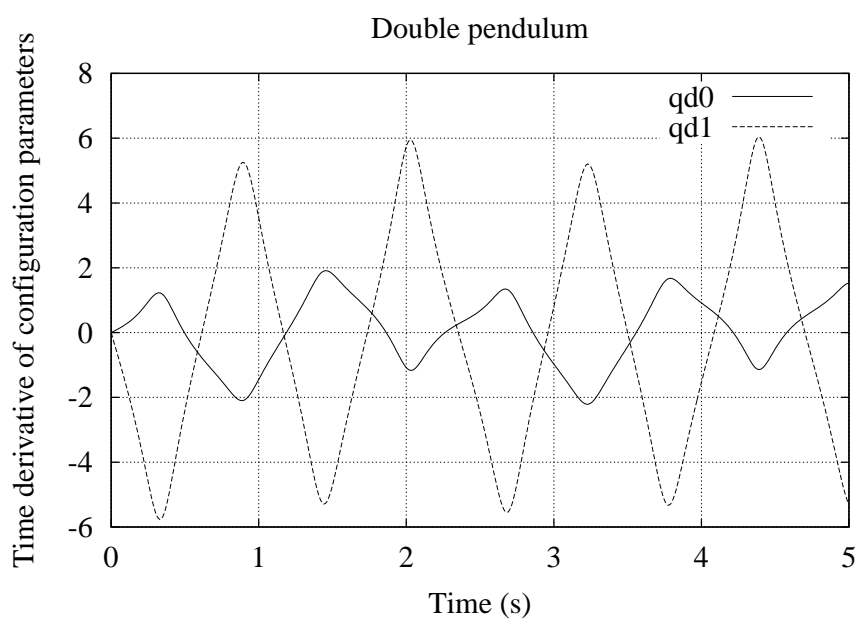


FIG. 2 – Evolution temporelle des dérivées premières des paramètres de configuration

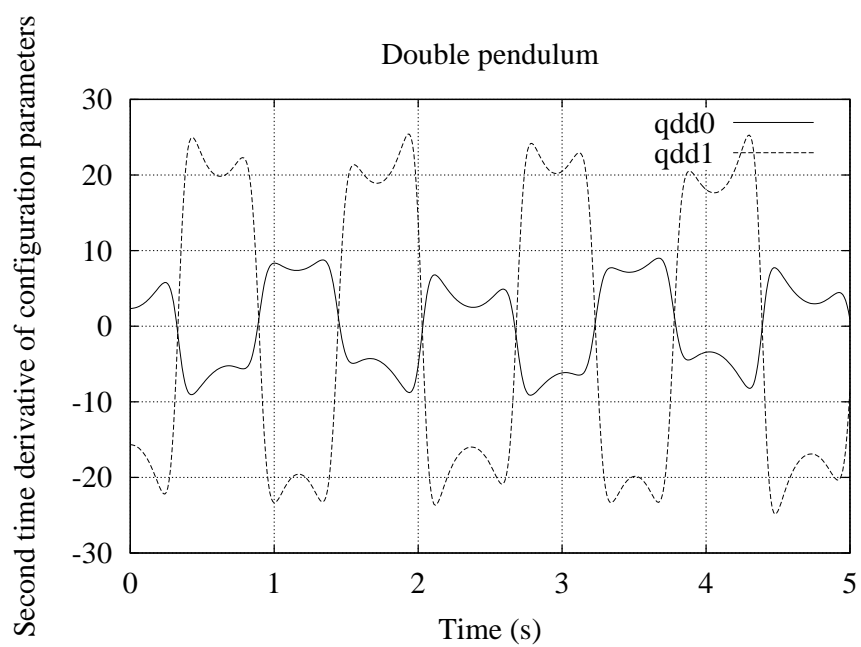


FIG. 3 – Evolution temporelle des dérivées secondes des paramètres de configuration

A Listing du fichier MuPAD

```
//

// 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 langage MuPAD)
//
//
//                               Ir. Georges KOUROUSSIS - mars 2003
//
//
// Title of the application
titre:="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:
nbrcont:= 0:

// 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
TempsFinal:=5:
StepSave:=0.01:
StepMax:=0.005:

// Uncomment if you want the LaTeX report
LATEX:=1
```