

Simulation of a slider-crank mechanism

27 octobre 2003

1 Données générales du mécanisme étudié

Le nombre de solides du système s'élève à 3 (défini par la variable globale `nbrbody`). Par la suite, ils se définissent par S_j avec j allant de 0 à 2. Le mécanisme analysé est desmodrome (à un degré de liberté) et donc un seul paramètre de configuration est défini par q_0 .

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.

$$m_{S0} = 1 \text{ kg}$$

$$m_{S1} = 2 \text{ kg}$$

$$m_{S2} = 5 \text{ kg}$$

$$\Phi_{G,S0} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.0833 \end{pmatrix}, \text{ en } kg.m^2$$

$$\Phi_{G,S1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.6667 \end{pmatrix}, \text{ en } kg.m^2$$

$$\Phi_{G,S2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \text{ en } kg.m^2$$

2 Paramètres cinématiques calculés sous MuPAD

Ces paramètres ont été calculés à partir du fichier utilisateur `slidercrank.mu` et ce, avec un temps de calcul *CPU* de 2 secondes.

Les matrices de transformation homogène pour chaque solide

$$T_{0G,S0} = \begin{pmatrix} \cos(q_0) & -\sin(q_0) & 0 & \frac{\cos(q_0)}{2} \\ \sin(q_0) & \cos(q_0) & 0 & \frac{\sin(q_0)}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{0G,S1} = \begin{pmatrix} \sqrt{\frac{\cos(2q_0)}{8} + 7/8} & \frac{\sin(q_0)}{2} & 0 & \cos(q_0) + \sqrt{\frac{\cos(2q_0)}{8} + 7/8} \\ -\frac{\sin(q_0)}{2} & \sqrt{\frac{\cos(2q_0)}{8} + 7/8} & 0 & \frac{\sin(q_0)}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{0G,S2} = \begin{pmatrix} 1 & 0 & 0 & \cos(q_0) + 2\sqrt{\frac{\cos(2q_0)}{8} + 7/8} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Leur dérivée par rapport au temps

$$\dot{T}_{0G,S0} = \begin{pmatrix} -\dot{q}_0 \sin(q_0) & -\dot{q}_0 \cos(q_0) & 0 & -\frac{\dot{q}_0 \sin(q_0)}{2} \\ \dot{q}_0 \cos(q_0) & -\dot{q}_0 \sin(q_0) & 0 & \frac{\dot{q}_0 \cos(q_0)}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\dot{T}_{0G,S1} = \begin{pmatrix} -\frac{\dot{q}_0 \sin(2q_0)}{8\sqrt{\frac{\cos(2q_0)}{8} + 7/8}} & \frac{\dot{q}_0 \cos(q_0)}{2} & 0 & \frac{-\frac{\dot{q}_0 \sin(2q_0)}{2} - 4\dot{q}_0 \sin(q_0)\sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{4\sqrt{\frac{\cos(2q_0)}{8} + 7/8}} \\ -\frac{\dot{q}_0 \cos(q_0)}{2} & -\frac{\dot{q}_0 \sin(2q_0)}{8\sqrt{\frac{\cos(2q_0)}{8} + 7/8}} & 0 & \frac{\dot{q}_0 \cos(q_0)}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\dot{T}_{0G,S2} = \begin{pmatrix} 0 & 0 & 0 & \frac{-\frac{\dot{q}_0 \sin(2q_0)}{2} - 2\dot{q}_0 \sin(q_0)\sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{2\sqrt{\frac{\cos(2q_0)}{8} + 7/8}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Les vitesses des centres de gravité des solides

$$\vec{v}_{G,S0} = \begin{pmatrix} -\frac{\dot{q}_0 \sin(q_0)}{2} \\ \frac{\dot{q}_0 \cos(q_0)}{2} \\ 0 \end{pmatrix}$$

$$\vec{v}_{G,S1} = \begin{pmatrix} \frac{-\frac{\dot{q}_0 \sin(2q_0)}{2} - 4\dot{q}_0 \sin(q_0) \sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{4\sqrt{\frac{\cos(2q_0)}{8} + 7/8}} \\ \frac{\dot{q}_0 \cos(q_0)}{2} \\ 0 \end{pmatrix}$$

$$\vec{v}_{G,S2} = \begin{pmatrix} \frac{-\frac{\dot{q}_0 \sin(2q_0)}{2} - 2\dot{q}_0 \sin(q_0) \sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{2\sqrt{\frac{\cos(2q_0)}{8} + 7/8}} \\ 0 \\ 0 \end{pmatrix}$$

Les accélérations des centres de gravité des solides

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

$$\vec{a}_{G,S1} = \begin{pmatrix} \frac{\dot{q}_0^2 \cos(4q_0) \sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{8} - \frac{27\ddot{q}_0 \sin(3q_0)}{16} - \frac{113\dot{q}_0^2 \cos(q_0)}{8} - \frac{\ddot{q}_0 \sin(5q_0)}{16} - \frac{29\dot{q}_0^2 \cos(3q_0)}{16} - \frac{\dot{q}_0^2 \cos(5q_0)}{16} - \frac{\dot{q}_0^2 \sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{8} - \frac{85\ddot{q}_0}{1} \\ \hline \end{pmatrix}$$

$$\vec{a}_{G,S2} = \begin{pmatrix} \frac{\dot{q}_0^2 \cos(4q_0) \sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{8} - \frac{27\ddot{q}_0 \sin(3q_0)}{32} - \frac{113\dot{q}_0^2 \cos(q_0)}{16} - \frac{\ddot{q}_0 \sin(5q_0)}{32} - \frac{29\dot{q}_0^2 \cos(3q_0)}{32} - \frac{\dot{q}_0^2 \cos(5q_0)}{32} - \frac{\dot{q}_0^2 \sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{8} - \frac{85\ddot{q}_0}{1} \\ \hline \end{pmatrix}$$

Les vitesses de rotation des solides

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

$$\vec{\omega}_{S1} = \begin{pmatrix} 0 \\ 0 \\ -\frac{\dot{q}_0 \cos(q_0)}{2\sqrt{\frac{\cos(2q_0)}{8} + 7/8}} \end{pmatrix}$$

$$\vec{\omega}_{S2} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Les accélérations de rotation des solides

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

$$\vec{\ddot{\omega}}_{S1} = \begin{pmatrix} \frac{\dot{q}_0^2 \sin(5q_0) \sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{16} - \frac{\dot{q}_0^2 \sin(3q_0) \sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{16} - \frac{\dot{q}_0^2 \sin(q_0) \sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{8} + \ddot{q}_0 \cos(q_0) \left(\frac{\sin(q_0)^2 \sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{8} - \frac{\cos(q_0)^2 \sqrt{\frac{\cos(2q_0)}{8} + 7/8}}{8} \right) \end{pmatrix}$$

$$\vec{\ddot{\omega}}_{S2} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

3 Simulation du mécanisme

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

- *FinalTime* est la durée de la simulation (=6 s),
- *StepSave* est le pas de temps dans l'intégration numérique (=0.05 s),
- *StepMax* est un pas de temps limite (=0.01 s),

les conditions initiales étant $q_0 = 1.0$, 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 Gnuplot comme nous le montrent les figures 1 à 3 créées par le code suivant :

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

set term postscript eps color "Times-Roman" 20
set output "figure1.eps"
set ylabel "displacements"
plot 'slidercrank.res' using 1:2 title 'q_0' 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 'slidercrank.res' using 1:3 title 'qd_0' 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 'slidercrank.res' using 1:4 title 'qdd_0' with line
set term X11
replot
```

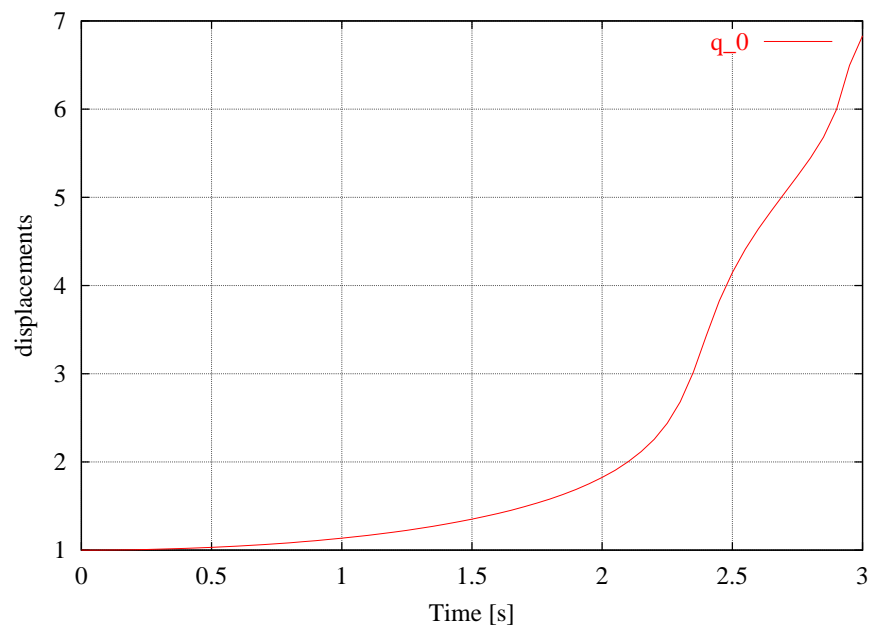


FIG. 1 – Evolution temporelle des paramètres de configuration

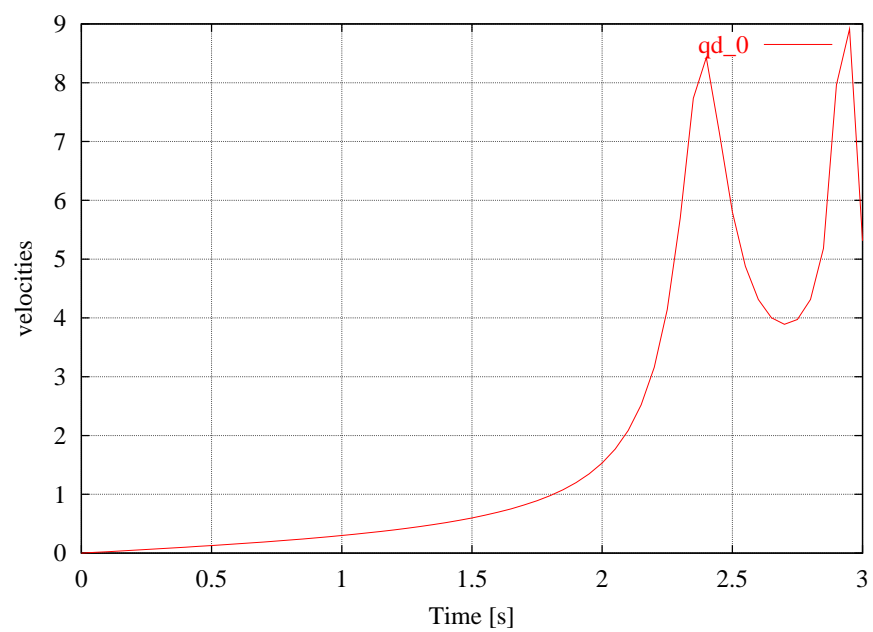


FIG. 2 – Evolution temporelle des dérivées premières des paramètres de configuration

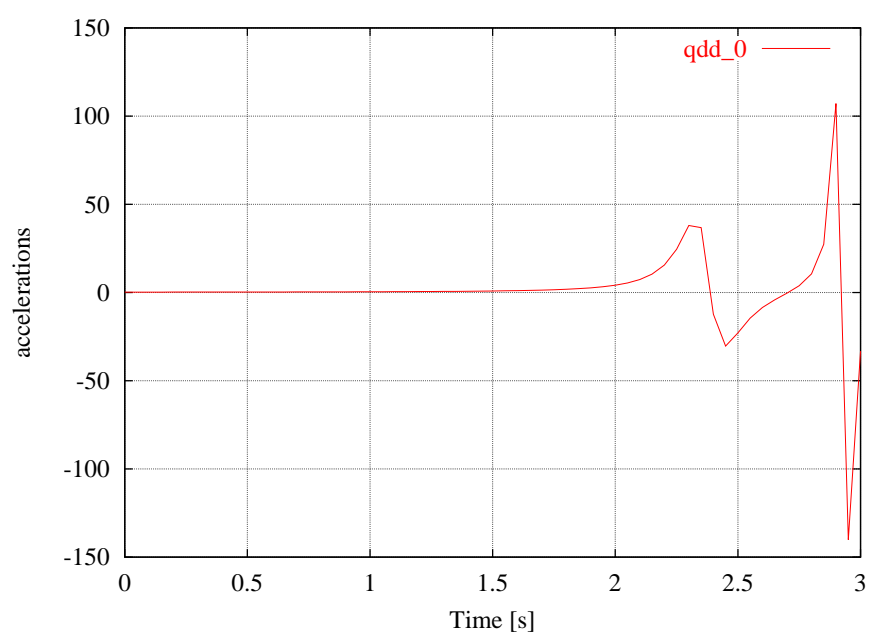


FIG. 3 – Evolution temporelle des dérivées secondes des paramètres de configuration

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 application
title:="Simulation of a slider-crank mechanism":

// Définition de nbrdof : Number of degrees of freedom,
//                               nbrbody : Number of bodies
nbrdof:= 1:
nbrbody:= 3:

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

// Inertia characteristics
mass[0]:=1:
mass[1]:=2:
mass[2]:=5:
Ixx[0]:=1:
Iyy[0]:=1:
Izz[0]:=0.0833:
Ixx[1]:=1:
```



```
Iyy[1]:=1:
Izz[1]:=0.6667:
Ixx[2]:=1:
Iyy[2]:=1:
Izz[2]:=2:

// Eventual constants
lc := 1: // length of the crank
lr := 2: // length of the rod

// Intermediary variables for the kinematics
alpha := arcsin(lc*sin(q[0])/lr):
x3 := lc*cos(q[0]) + lr*cos(alpha):

// Definition of the homogeneous transformation matrices
TOG[0]:= Trotz(q[0]) * Tdisp(lc/2,0,0):
TOG[1] := Tdisp(x3,0,0) * Trotz(-alpha) * Tdisp(-lr/2,0,0):
TOG[2] := Tdisp(x3,0,0):

// Initial conditions
qi[0]:=1.0:
qdi[0]:=0:

// Simulation parameters
FinalTime:=6:
StepSave:=0.05:
StepMax:=0.01:

SIMPLIFY:=1:
// Set FORCES to 1 in case you want to include *.AppEff.cpp into procedure
// AddAppliedEfforts() to define forces other than gravity
FORCES:=1:
// 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:
```