Motivating example (MIPSI): Babyboot

Model
The figure to the right is a schematic representation of a swinging babyboot attached by a shoelace to a rigid support. The mechanical model of the babyboot consists of a thin uniform rod $A$ attached to a fixed support $N$ by a revolute joint, and a uniform plate $B$ connected to $A$ with a second revolute joint so that $B$ can rotate freely about $A$’s axis. Note: The revolute joints’ axes are perpendicular, not parallel.

- **Bodies**: The rod and plate are rigid (inflexible/undeformable).
- **Connections**: The revolute joints are ideal (massless, frictionless, with no slop or flexibility).
- **Force**: Earth’s gravity is uniform and constant.

Other contact forces (e.g., air resistance, solar/light pressure) and distance forces (e.g., electromagnetic, other gravitational) are negligible.

- **Newtonian reference frame**: Earth

Identifiers
Right-handed sets of unit vectors $\hat{n}_x, \hat{n}_y, \hat{n}_z$: $\hat{a}_x, \hat{a}_y, \hat{a}_z$: and $\hat{b}_x, \hat{b}_y, \hat{b}_z$ are fixed in $N$, $A$, and $B$, respectively, with $\hat{n}_x = \hat{a}_x$ parallel to the revolute axis joining $A$ to $N$, $\hat{n}_z$ vertically upward, $\hat{a}_z = \hat{b}_z$ parallel to the rod’s long axis (and the revolute axis joining $B$ to $A$), and $\hat{b}_z$ perpendicular to plate $B$.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Symbol</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earth’s gravitational constant</td>
<td>$g$</td>
<td>Constant</td>
<td>$9.81 \text{ m/s}^2$</td>
</tr>
<tr>
<td>Distance between $N_o$ and $A_{cm}$</td>
<td>$L_A$</td>
<td>Constant</td>
<td>$7.5 \text{ cm}$</td>
</tr>
<tr>
<td>Distance between $N_o$ and $B_{cm}$</td>
<td>$L_B$</td>
<td>Constant</td>
<td>$20 \text{ cm}$</td>
</tr>
<tr>
<td>Mass of $A$</td>
<td>$m^A$</td>
<td>Constant</td>
<td>$0.01 \text{ kg}$</td>
</tr>
<tr>
<td>Mass of $B$</td>
<td>$m^B$</td>
<td>Constant</td>
<td>$0.1 \text{ kg}$</td>
</tr>
<tr>
<td>$A$’s moment of inertia about $A_{cm}$ for $\hat{a}_x$</td>
<td>$I^A_x$</td>
<td>Constant</td>
<td>$0.05 \text{ kg*cm}^2$</td>
</tr>
<tr>
<td>$B$’s moment of inertia about $B_{cm}$ for $\hat{b}_x$</td>
<td>$I^B_y$</td>
<td>Constant</td>
<td>$2.5 \text{ kg*cm}^2$</td>
</tr>
<tr>
<td>$B$’s moment of inertia about $B_{cm}$ for $\hat{b}_y$</td>
<td>$I^B_z$</td>
<td>Constant</td>
<td>$0.5 \text{ kg*cm}^2$</td>
</tr>
<tr>
<td>$B$’s moment of inertia about $B_{cm}$ for $\hat{b}_z$</td>
<td>$I^B_z$</td>
<td>Constant</td>
<td>$2.0 \text{ kg*cm}^2$</td>
</tr>
<tr>
<td>Angle from $\hat{n}_z$ to $\hat{a}_x$ with $\hat{n}_x$ sense</td>
<td>$q_A$</td>
<td>Dependent variable</td>
<td>Varies</td>
</tr>
<tr>
<td>Angle from $\hat{a}_y$ to $\hat{b}_y$ with $\hat{a}_z$ sense</td>
<td>$q_B$</td>
<td>Dependent variable</td>
<td>Varies</td>
</tr>
<tr>
<td>Time</td>
<td>$t$</td>
<td>Independent variable</td>
<td>Varies</td>
</tr>
</tbody>
</table>

Physics

The ODEs (ordinary differential equations) governing the motion of this mechanical system are\(^\text{11}\)

$$\ddot{q}_A = \frac{2 \dot{q}_A \dot{q}_B \sin(q_B) \cos(q_B)}{I^A + m^A L_A^2 + m^B L_B^2} \bigg( I^B_x - I^B_y \bigg) - \frac{m^A L_A m^B L_B g \sin(q_A)}{I^A + m^A L_A^2 + m^B L_B^2 \cos^2(q_B) + I^B_y \sin^2(q_B)}$$

$$\ddot{q}_B = \frac{-\dot{q}_A^2 \sin(q_B) \cos(q_B) \big(I^B_x - I^B_y\big)}{I^B_x}$$

\(^\text{11}\)Four methods for forming equations of motion are: Free-body diagrams of $A$ and $B$ (which is inefficient as it introduces up to 10 unknown force/torque measures); D’Alembert’s method (road maps of Section 22.6) which efficiently forms the two equations shown for $\dot{q}_A$ and $\dot{q}_B$ (but require a clever selection of systems, points, and unit vectors); Lagrange’s equations (an energy-based method that automates D’Alembert’s cleverness); Kane’s equations (a modern efficient blend of D’Alembert and Lagrange).

Copyright © 2009-2015 Motion Genesis LLC. www.MotionGenesis.com
Simplify and solve

Computers have revolutionized the solution of differential equations. There are many numerical algorithms for solving nonlinear, coupled, variable coefficient, ODEs (ordinary differential equations) including Euler's method, predictor-corrector, Runge-Kutta, etc. In addition, there are many programs (MATLAB®, MotionGenesis, WolframAlpha, etc.) that make it easy to solve ODEs.

Computer (numerical) solution of ODEs with MotionGenesis (with plotting)

Variable $q^\prime\prime A$, $q^\prime\prime B$  % Angles and first/second time-derivatives.

\[
q^\prime\prime A = \frac{2 \times (508.89 \times \sin(qA) - \sin(qB) \times \cos(qB) \times qA' \times qB')}{(-21.556 + \sin(qB)^2)}
\]

\[
q^\prime\prime B = -\sin(qB) \times \cos(qB) \times qA'^2
\]

Input $t_{\text{Final}} = 10 \text{ sec}$, $t_{\text{Step}} = 0.02 \text{ sec}$, $\text{absError} = 1.0 \times 10^{-7}$

Input $qA = 90 \ deg$, $qB = 1.0 \ deg$, $qA' = 0.0 \ rad/sec$, $qB' = 0.0 \ rad/sec$

OutputPlot $t \ sec$, $qA \ degrees$, $qB \ degrees$

ODE() solveBabybootODE

Quit

Interpret

The solution to these differential equations reveals this simple system has strange, non-intuitive motion.\textsuperscript{12} For certain initial values of $qA$, the motion of plate B is well-behaved and “stable”. Alternately, for other initial values of $qA$, B’s motion is “chaotic” – meaning that a small variation in the initial value of $qB$ or numerical integration inaccuracies lead to dramatically different results (these ODEs are used to test the accuracy of numerical integrators – the plots below required a numerical integrator error of $\text{absError} = 1 \times 10^{-7}$).

The following chart and figure to the right shows this system’s regions of stability (black) and instability (green). Notice the “chaotic” plot below shows $qB$ is very sensitive to initial values. A $0.5^\circ$ change in the initial value of $qB(0)$ results in more than a $2000^\circ$ difference in the value of $qB(t = 10)$!

<table>
<thead>
<tr>
<th>Initial value of $qA$</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0^\circ \leq qA(0) \leq 71.3^\circ$</td>
<td>Stable black</td>
</tr>
<tr>
<td>$71.4^\circ \leq qA(0) \leq 111.77^\circ$</td>
<td>Unstable green</td>
</tr>
<tr>
<td>$111.78^\circ \leq qA(0) \leq 159.9^\circ$</td>
<td>Stable black</td>
</tr>
<tr>
<td>$160.0^\circ \leq qA(0) \leq 180.0^\circ$</td>
<td>Unstable green</td>
</tr>
</tbody>
</table>

Investigation of stability: More simulation results

<table>
<thead>
<tr>
<th>Stable: Released from 5°</th>
<th>Stable: Released from 30°</th>
<th>Chaotic: Released from 90°</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Graph 1" /></td>
<td><img src="image2.png" alt="Graph 2" /></td>
<td><img src="image3.png" alt="Graph 3" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stable: Released from 135°</th>
<th>Beat: Released from 158°</th>
<th>Chaotic: Released from 177°</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image4.png" alt="Graph 4" /></td>
<td><img src="image5.png" alt="Graph 5" /></td>
<td><img src="image6.png" alt="Graph 6" /></td>
</tr>
</tbody>
</table>

**Physics**


```plaintext
\% File: BabybootWithDAlembertMethod.txt
\% Problem: Analysis of 3D chaotic double pendulum
\% Copyright (c) 2009 Motion Genesis LLC. All rights reserved.
\%----------------------------------------------------------------
SetDigits( 5 ) \% Number of digits displayed for numbers
\%----------------------------------------------------------------
NewtonianFrame N
RigidBody A \% Upper rod
RigidBody B \% Lower plate
\%----------------------------------------------------------------
Variable qA'' \% Pendulum angle and its time-derivatives
Variable qB'' \% Plate angle and its time-derivative
Constant LA = 7.5 cm \% Distance from pivot to A's mass center
Constant LB = 20 cm \% Distance from pivot to B's mass center
A.SetMassInertia( mA = 10 grams, IAx = 50 g*cm^2, IAy, IAz )
B.SetMassInertia( mB = 100 grams, IBx = 2500 g*cm^2, IBy = 500 g*cm^2, IBz = 2000 g*cm^2 )
\%----------------------------------------------------------------
% Rotational and translational kinematics
A.RotateX( N, qA )
B.RotateZ( A, qB )
Acm.Translate( No, -LA*Az> )
Bcm.Translate( No, -LB*Az> )
\%----------------------------------------------------------------
% Add relevant forces
\% g> = -9.81*Nz>
System.AddForceGravity( g> )
\%----------------------------------------------------------------
% Rotational equations of motion for B and A+B.
Dynamics[1] = Dot( B.GetDynamics(Bcm), Bz> )
Dynamics[2] = Dot( System(A,B).GetDynamics(No), Ax> )
Solver( Dynamics, qA'', qB'' )
\%----------------------------------------------------------------
% Kinetic and potential energy
KE = System.GetKineticEnergy()
PE = System.GetForceGravityPotentialEnergy( g>, No )
Energy = KE + PE
\%----------------------------------------------------------------
% Integration parameters and initial values.
Input tFinal=10, tStep=0.02, absError=1.0E-07, relError=1.0E-07
Input qA = 90 deg, qA' = 0.0 rad/sec, qB = 1.0 deg, qB' = 0.0 rad/sec
\%----------------------------------------------------------------
% List output quantities and solve ODEs.
OutputPlot t sec, qA deg, qB deg, Energy N*m
ODE() BabybootDAlembert
\%----------------------------------------------------------------
Save BabybootWithDAlembertMethod.all
Quit
```

Copyright © 2009-2015 Motion Genesis LLC.