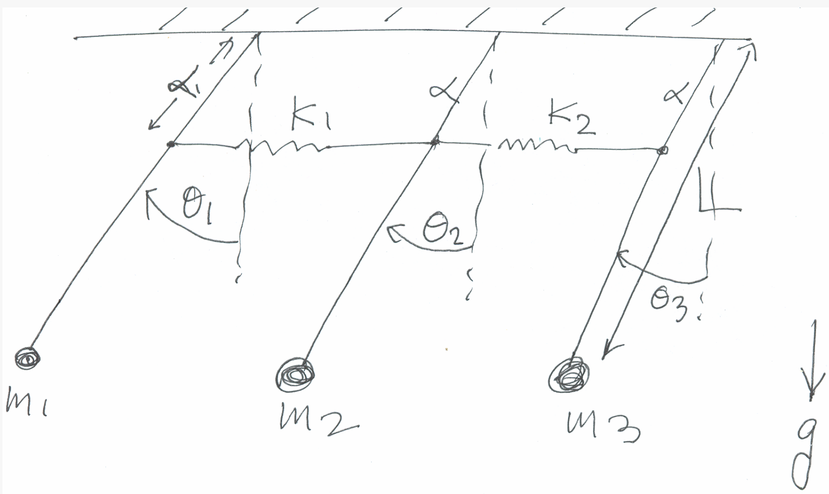
Modal Analysis for 3 pendulum with springs problem

by Nasser M Abbasi

This small note shows how to use Mathematica to solve symbolically for a problem in vibration to find the natural frequencies that a system of 3 masses will vibrate in. This digram below describes the problem. We use Lagrangian formulation to determine the equation of motions, then use modal analysis to decouple the system and solve it. In this system , the springs are attached at a distant α From the edge. Each pendulum has length L and has masses m1,m2,m3 attached to the end. The initial conditions are θ(0)={Pi/4,Pi/4,Pi/4} and $\dot{\theta}$={0,0,0}



Define a function which accepts the kinetic and potential energy and return back the stiffness and the mass matrix

In[1]:=
```
Needs["Notation`"]
```

In[2]:=
```
Symbolize[θ₁];
Symbolize[θ₂];
Symbolize[θ₃];
Symbolize[r₁];
Symbolize[r₂];
Symbolize[r₃];
```

In[8]:=
```
obtainStiffnessAndMassMatrix[ke_, u_] :=
 Module[{lagrangian, eq1, eq2, eq3, meq1, meq2, meq3,
   seq1, seq2, seq3, stiffnessMatrix, massMatrix, any},
  lagrangian = ke - u;
  eq1 = D[D[lagrangian, θ1'[t]], t] - D[lagrangian, θ1[t]] == 0;
  eq2 = D[D[lagrangian, θ2'[t]], t] - D[lagrangian, θ2[t]] == 0;
  eq3 = D[D[lagrangian, θ3'[t]], t] - D[lagrangian, θ3[t]] == 0;

  (*linearize using small angle approximation*)
  eq1 = eq1 /. Sin[θ1[t]] -> θ1[t];
  eq2 = eq2 /. Sin[θ2[t]] -> θ2[t];
  eq3 = eq3 /. Sin[θ3[t]] -> θ3[t];

  Print["Equation of motion are"];
  Print[{eq1, eq2, eq3} // TableForm];

  seq1 = eq1 /. θ1''[t] _ → 0;
  seq2 = eq2 /. θ2''[t] _ → 0;
  seq3 = eq3 /. θ3''[t] _ → 0;

  z1 = eq1 /. any_ == 0 → any;
  z2 = seq1 /. any_ == 0 → any;
  meq1 = (eq1 /. any_ == 0 → any) - (seq1 /. any_ == 0 → any) // Simplify;

  z1 = eq2 /. any_ == 0 → any;
  z2 = seq2 /. any_ == 0 → any;
  meq2 = (z1 - z2) // FullSimplify;

  meq3 = (eq3 /. any_ == 0 → any) - (seq3 /. any_ == 0 → any) // Simplify;

  r = Normal[CoefficientArrays[{seq1, seq2, seq3}, {θ1[t], θ2[t], θ3[t]}]];
  stiffnessMatrix = Collect[r[[2, All]], α];
  r = Normal[CoefficientArrays[{meq1, meq2, meq3}, {θ1''[t], θ2''[t], θ3''[t]}]];
  massMatrix = Collect[r[[2, All]], α];
  {massMatrix, stiffnessMatrix}
 ]
```

Now define the kinetic and potenatial energy

In[9]:=
$$ke = \frac{1}{2} m_1 (L\,\theta_1'[t])^2 + \frac{1}{2} m_2 (L\,\theta_2'[t])^2 + \frac{1}{2} m_3 (L\,\theta_3'[t])^2;$$
$$u = \frac{1}{2} k_1 (\alpha\,\theta_2[t] - \alpha\,\theta_1[t])^2 + \frac{1}{2} k_2 (\alpha\,\theta_3[t] - \alpha\,\theta_2[t])^2 + m_1 g L (1 - \mathrm{Cos}[\theta_1[t]]) + m_2 g L (1 - \mathrm{Cos}[\theta_2[t]]) + m_3 g L (1 - \mathrm{Cos}[\theta_3[t]]);$$

Now call the above function to generate the stiffness and mass matrix. It also prints the 3 equations of motion

In[11]:=
```
{massMatrix, stiffnessMatrix} = obtainStiffnessAndMassMatrix[ke, u];
```

Equation of motion are

$$g L m_1 \theta_1[t] - \alpha k_1 (-\alpha \theta_1[t] + \alpha \theta_2[t]) + L^2 m_1 (\theta_1)''[t] == 0$$
$$g L m_2 \theta_2[t] + \alpha k_1 (-\alpha \theta_1[t] + \alpha \theta_2[t]) - \alpha k_2 (-\alpha \theta_2[t] + \alpha \theta_3[t]) + L^2 m_2 (\theta_2)''[t] == 0$$
$$g L m_3 \theta_3[t] + \alpha k_2 (-\alpha \theta_2[t] + \alpha \theta_3[t]) + L^2 m_3 (\theta_3)''[t] == 0$$

Now print the STIFFNESS and MASS matrix

In[12]:=
```
vars = {θ1''[t], θ2''[t], θ3''[t]};
deps = {θ1[t], θ2[t], θ3[t]};
Print[MatrixForm[massMatrix], MatrixForm[Transpose[List[vars]]], "+",
  MatrixForm[stiffnessMatrix], MatrixForm[Transpose[List[deps]]]];
```

$$\begin{pmatrix} L^2 m_1 & 0 & 0 \\ 0 & L^2 m_2 & 0 \\ 0 & 0 & L^2 m_3 \end{pmatrix} \begin{pmatrix} (\theta_1)''[t] \\ (\theta_2)''[t] \\ (\theta_3)''[t] \end{pmatrix} + \begin{pmatrix} \alpha^2 k_1 + g L m_1 & -\alpha^2 k_1 & 0 \\ -\alpha^2 k_1 & \alpha^2 (k_1 + k_2) + g L m_2 & -\alpha^2 k_2 \\ 0 & -\alpha^2 k_2 & \alpha^2 k_2 + g L m_3 \end{pmatrix} \begin{pmatrix} \theta_1[t] \\ \theta_2[t] \\ \theta_3[t] \end{pmatrix}$$

Now that we have the stiffness and mass matrix, we can perform modal analysis. Start by doing the first transformation

In[16]:=
```
invMassMatrix = MatrixPower[massMatrix, -1/2];
Print["new K matrix= ",
  MatrixForm[newK = invMassMatrix .stiffnessMatrix .invMassMatrix]];
```

new K matrix= $\begin{pmatrix} \frac{\alpha^2 k_1 + g L m_1}{L^2 m_1} & -\frac{\alpha^2 k_1}{\sqrt{L^2 m_1}\sqrt{L^2 m_2}} & 0 \\ -\frac{\alpha^2 k_1}{\sqrt{L^2 m_1}\sqrt{L^2 m_2}} & \frac{\alpha^2 (k_1+k_2) + g L m_2}{L^2 m_2} & -\frac{\alpha^2 k_2}{\sqrt{L^2 m_2}\sqrt{L^2 m_3}} \\ 0 & -\frac{\alpha^2 k_2}{\sqrt{L^2 m_2}\sqrt{L^2 m_3}} & \frac{\alpha^2 k_2 + g L m_3}{L^2 m_3} \end{pmatrix}$

In[20]:=
```
values = {m1 → 1, m2 → 2, m3 → 3, k1 → 10, L → 1, α → .5, k2 → 2, g → 9.8};
invMassMatrix = invMassMatrix /. values;
massMatrix = massMatrix /. values;
newK = newK /. values
```

Out[23]=
```
{{12.3, -1.76777, 0}, {-1.76777, 11.3, -0.204124}, {0, -0.204124, 9.96667}}
```

In[24]:=
```
{lambda, eigvect} = Eigensystem[newK];
Print["Eigenvalues=", lambda];
```

Eigenvalues={13.6413, 10.1254, 9.8}

In[26]:=
```
eigvect = Transpose[eigvect];
eigvect = eigvect / Map[Norm, eigvect];
Print["Eigenvectors=", eigvect];
```

Eigenvectors=
{{0.796202, -0.446538, 0.408248}, {-0.6041, -0.5493, 0.57735}, {0.0335579, 0.70631, 0.707107}}

Now find Λ matrix

In[29]:=
```
Λ = Transpose[eigvect].newK.eigvect // Chop
```

Out[29]=
```
{{13.6413, 0, 0}, {0, 10.1254, 0}, {0, 0, 9.8}}
```

Hence the decoupled system of differential equations is

In[30]:=
```
vars = {r1''[t], r2''[t], r3''[t]};
deps = {r1[t], r2[t], r3[t]};
Print[MatrixForm[Transpose[List[vars]]], "+",
  MatrixForm[Λ], MatrixForm[Transpose[List[deps]]]];
```

$$\begin{pmatrix} (r_1)''[t] \\ (r_2)''[t] \\ (r_3)''[t] \end{pmatrix} + \begin{pmatrix} 13.6413 & 0 & 0 \\ 0 & 10.1254 & 0 \\ 0 & 0 & 9.8 \end{pmatrix} \begin{pmatrix} r_1[t] \\ r_2[t] \\ r_3[t] \end{pmatrix}$$

Now convert the IC from *θ* (t) space to r (t) space

In[35]:=
```
initialθ = {Pi / 4, Pi / 2, Pi / 4};
initialR = Transpose[eigvect].MatrixPower[massMatrix, 1/2].initialθ // Chop
```

Out[36]=
```
{-0.670986, -0.610119, 2.5651}
```

In[37]:=
```
initialθDot = {0, 0, 0};
initialRDot = Transpose[eigvect].MatrixPower[massMatrix, 1/2].initialθDot
```

Out[38]=
```
{0., 0., 0.}
```

Now solve the r (t) system

In[42]:=
```
eq = r1''[t] + Λ[[1, 1]] r1[t] == 0;
solr1 =
  First[DSolve[{eq, r1[0] == initialR[[1]], r1'[0] == initialRDot[[1]]}, r1[t], t]];
Print["r1[t]=", solr1 = r1[t] /. solr1]
```

$r_1[t]=-0.670986\ Cos[3.69341\ t]$

In[45]:=
```
eq = r2''[t] + Λ[[2, 2]] r2[t] == 0;
solr2 =
  First[DSolve[{eq, r2[0] == initialR[[2]], r2'[0] == initialRDot[[2]]}, r2[t], t]];
Print["r2[t]=", solr2 = r2[t] /. solr2]
```

$r_2[t]=-0.610119\ Cos[3.18205\ t]$

In[48]:=
```
eq = r3''[t] + Λ[[3, 3]] r3[t] == 0;
solr3 =
  First[DSolve[{eq, r3[0] == initialR[[3]], r3'[0] == initialRDot[[3]]}, r3[t], t]];
Print["r3[t]=", solr3 = r3[t] /. solr3]
```

$r_3[t]$=2.5651 Cos[3.1305 t]

Now convert solution from r (t) to *θ* (t)

In[51]:=
```
MatrixForm[solθ = invMassMatrix.eigvect.{{solr1}, {solr2}, {solr3}}]
```

Out[51]//MatrixForm=

$$\begin{pmatrix} 1.0472\,\mathrm{Cos}[3.1305\,t] + 0.272441\,\mathrm{Cos}[3.18205\,t] - 0.53424\,\mathrm{Cos}[3.69341\,t] \\ 1.0472\,\mathrm{Cos}[3.1305\,t] + 0.236978\,\mathrm{Cos}[3.18205\,t] + 0.28662\,\mathrm{Cos}[3.69341\,t] \\ 1.0472\,\mathrm{Cos}[3.1305\,t] - 0.248799\,\mathrm{Cos}[3.18205\,t] - 0.0130001\,\mathrm{Cos}[3.69341\,t] \end{pmatrix}$$

In[52]:=
```
solθ1 = solθ[[1]];
solθ2 = solθ[[2]];
solθ3 = solθ[[3]];
```

Now plot the solutions

In[55]:=
```
Print["θ1[t]=", solθ1 = First[Simplify[Chop[ExpToTrig[solθ1 /. values]]]]];
```

$\theta_1[t]$=1.0472 Cos[3.1305 t] + 0.272441 Cos[3.18205 t] - 0.53424 Cos[3.69341 t]

In[56]:=
```
Print["θ2[t]=", solθ2 = First[FullSimplify[Chop[ExpToTrig[solθ2 /. values]]]]];
```

$\theta_2[t]$=1.0472 Cos[3.1305 t] + 0.236978 Cos[3.18205 t] + 0.28662 Cos[3.69341 t]

In[57]:=
```
Print["θ3[t]=", solθ3 = First[Simplify[Chop[ExpToTrig[solθ3 /. values]]]]];
```

$\theta_3[t]$=1.0472 Cos[3.1305 t] - 0.248799 Cos[3.18205 t] - 0.0130001 Cos[3.69341 t]

In[58]:=
```
Plot[{solθ1, solθ2, solθ3}, {t, 0, 10}]
```

Out[58]=