In[55]:=
```
(*version 5/30/2015, copyright by Nasser M. Abbasi*)
Manipulate[

 tick;

 Module[{debug = False, eq, r1 = 2, r2 = 1,
   r3 = .8, L1 = 5, L2 = 5, L3 = 5, cq1, cq2, cq3, t, k1, k2, k3, k4},
  If[state == "RUN" || state == "STEP",

   k1 = zDot[{q1, q2, q3, dq1, dq2, dq3}, c1, c2, c3,
     m1, m2, m3, L1, L2, L3, r1, r2, r3, torque1, torque2, torque3];
   k2 = zDot[{q1, q2, q3, dq1, dq2, dq3} + 0.5 * k1 * delT, c1, c2, c3, m1,
     m2, m3, L1, L2, L3, r1, r2, r3, torque1, torque2, torque3];
   k3 = zDot[{q1, q2, q3, dq1, dq2, dq3} + 0.5 * k2 * delT, c1, c2, c3, m1,
     m2, m3, L1, L2, L3, r1, r2, r3, torque1, torque2, torque3];
   k4 = zDot[{q1, q2, q3, dq1, dq2, dq3} + k3 * delT, c1, c2, c3, m1,
     m2, m3, L1, L2, L3, r1, r2, r3, torque1, torque2, torque3];
   {q1, q2, q3, dq1, dq2, dq3} = {q1, q2, q3, dq1, dq2, dq3} +
     (1 / 6) * (k1 + 2 * k2 + 2 * k3 + k4) * delT;

   cq1 = q1;
   cq2 = q2;
   cq3 = q3;
   vEnd = getV3[L1, L2, L3, q1, q2, q3, dq1, dq2, dq3];

   (*Print["after setting q1=",q1];*)

   ct = Mod[ct + delT, 1000];
   If[state == "RUN",
    tick = Not[tick]
   ]
   ,
   cq1 = q1;
   cq2 = q2;
   cq3 = q3
  ];

  g = Grid[{
     (*{Grid[{{q1,q2,q3,dq1,dq2,dq3}},Frame→All]},*)
     {
      Deploy@Graphics3D[
        {
```

```
          (*first link*)
          Rotate[
           {
            {LightGray, Cylinder[{{0, 0, -0.3 r1}, {0, 0, 0}}, 2 r1]},
            typeOne[r1, {0, 0, 0}, L1, False, True],
            Rotate[
             {typeOne[r2, {0, 0, L1}, L2, True, True],
              Rotate[
               {typeOne[r2, {0, 0, L1 + L2}, L3, True, False]
                (*{Red,Arrowheads[Small],
                 Arrow[{{0,0,L1+L2+L3},({0,0,L1+L2+L3}+vEnd/(.85L3))}]}*)
               },
               cq3, {1, 0, 0}, {0, 0, L1 + L2 + 0.2 r3}
              ]
             }, cq2, {1, 0, 0}, {0, 0, L1 + 0.2 r1}
            ]
           }, cq1, {0, 0, 1}
          ]

         },
         PlotRange → {{-L1 - 1.2 L2, L1 + 1.2 L2},
           {-L2 - 1.2 L3, L2 + 1.2 L3}, {-L2 - L3, L1 + L2 + 1.5 L3}},
         ImageSize → 300, Boxed → False, Axes → False, SphericalRegion → True,
         ViewPoint → {0, 1, 1}, Method → {"RotationControl" → None}
        ]
       },
       {Grid[
         {
          {"time", "θ1", "θ2", "θ3"},
          {padIt2[ct, {5, 2}], padIt2[Mod[q1 * 180. / Pi, 360], {3, 0}], padIt2[
            Mod[q2 * 180. / Pi, 360], {3, 0}], padIt2[Mod[q3 * 180. / Pi, 360], {3, 0}]}
         }, Frame → All]
       }
      }
     ];
    (*FinishDynamic[];*)
    g

   ],

   Text@Style[Grid[{
       {
```

```
 Grid[{
   {Button[Text@Style["run", 12], {state = "RUN";
      tick = Not[tick]}, ImageSize → {50, 40}],
    Button[Text@Style["step", 12], {state = "STEP";
      tick = Not[tick]}, ImageSize → {50, 40}],
    Button[Text@Style["stop", 12], {state = "STOP";
      tick = Not[tick]}, ImageSize → {50, 40}],
    Button[Text@Style["reset", 12], {state = "STEP";
      ct = 0;
      dq1 = 0.5;
      dq2 = 1;
      dq3 = 1;
      q1 = angle1;
      q2 = angle2;
      q3 = angle3;
      tick = Not[tick]}, ImageSize → {50, 40}]
   }}, Spacings → {1, 1}
 ], SpanFromLeft
},
{
 Grid[{{"Link 1 properties"},
   {"mass (kg)", Manipulator[Dynamic[m1, {m1 = #;
        tick = Not[tick]} &], {1, 100, 1},
     ImageSize → Tiny], Dynamic[padIt2[m1, 4]]},
   {"damping (Kg per sec)", Manipulator[Dynamic[c1, {c1 = #;
        tick = Not[tick]} &], {0, 1000, 1},
     ImageSize → Tiny], Dynamic[padIt2[c1, 4]]},
   {"initial angle", Manipulator[Dynamic[angle1, {angle1 = #;
        q1 = angle1 * Pi / 180.;
        state = "STOP";
        tick = Not[tick]} &], {0, 360, 1},
     ImageSize → Tiny], Dynamic[angle1]},
   {"applied joint torque", Manipulator[Dynamic[torque1, {torque1 = #;
        tick = Not[tick]} &], {-600, 600, 1},
     ImageSize → Tiny], Dynamic[padIt1[torque1, 3]],
    Spacer[2], Button[Text@Style["zero", 10], {torque1 = 0;
      tick = Not[tick]}, ImageSize → {40, 40}]
    , SpanFromLeft}
  }, Frame → True]
}
,
{
 Grid[{{"Link 2 properties"},
```

```
    {"mass (kg)",
     Manipulator[Dynamic[m2, {m2 = #; tick = Not[tick]} &], {1, 100, 1},
      ImageSize → Tiny], Dynamic[padIt2[m2, 4]]},
    {"damping (Kg per sec)", Manipulator[Dynamic[c2, {c2 = #;
          tick = Not[tick]} &], {0, 1000, 1},
      ImageSize → Tiny], Dynamic[padIt2[c2, 4]]},
    {"initial angle", Manipulator[Dynamic[angle2, {angle2 = #;
          q2 = angle2 * Pi / 180.;
          state = "STOP";
          tick = Not[tick]} &], {0, 165, 1},
      ImageSize → Tiny], Dynamic[angle2]},
    {"applied joint torque", Manipulator[Dynamic[torque2, {torque2 = #;
          tick = Not[tick]} &], {-600, 600, 1},
      ImageSize → Tiny], Dynamic[padIt1[torque2, 3]],
     Spacer[2], Button[Text@Style["zero", 10], {torque2 = 0;
       tick = Not[tick]}, ImageSize → {40, 40}],
     SpanFromLeft}
   }, Frame → True]
},
{
 Grid[{{"Link 3 properties"},
    {"mass (kg)", Manipulator[Dynamic[m3, {m3 = #;
          tick = Not[tick]} &], {1, 100, 1},
      ImageSize → Tiny], Dynamic[padIt2[m3, 4]]},
    {"damping (Kg per sec)", Manipulator[Dynamic[c3, {c3 = #;
          tick = Not[tick]} &], {0, 1000, 1},
      ImageSize → Tiny], Dynamic[padIt2[c3, 4]]},
    {"initial angle", Manipulator[Dynamic[angle3, {angle3 = #;
          q3 = angle3 * Pi / 180.;
          state = "STOP";
          tick = Not[tick]} &], {-75, 75, 1},
      ImageSize → Tiny], Dynamic[angle3]},
    {"applied joint torque", Manipulator[Dynamic[torque3, {torque3 = #;
          tick = Not[tick]} &], {-600, 600, 1},
      ImageSize → Tiny], Dynamic[padIt1[torque3, 3]],
     Spacer[2], Button[Text@Style["zero", 10], {torque3 = 0;
       tick = Not[tick]}, ImageSize → {40, 40}], SpanFromLeft}
   }, Frame → True]
},
{
 Grid[{{"simulation speed",
    Manipulator[Dynamic[delT, {delT = #;
          tick = Not[tick]} &], {0.001, 0.05, .001},
```

```
          ImageSize → Tiny], Dynamic[padIt2[delT, {2, 2}]]
        }}]
     }
    }, Frame → True, Alignment → Center, Spacings → {1, 1}
   ], 14],
{{g, 0}, None}, (*graphics*)
{{state, "STOP"}, None},
{{ct, 0}, None},
{{q1, 45 Degree}, None},
{{q2, 45 Degree}, None},
{{q3, 45 Degree}, None},
{{dq1, 0.5}, None},
{{dq2, 0.5}, None},
{{dq3, 0.5}, None},
{{delT, 0.02}, None},
{{angle1, 45}, None},
{{angle2, 45}, None},
{{angle3, 45}, None},
{{m1, 100}, None},
{{m2, 100}, None},
{{m3, 100}, None},
{{c1, 10}, None},
{{c2, 1}, None},
{{c3, 1}, None},
{{zoom, 10}, None},
{{torque1, 0}, None},
{{torque2, 0}, None},
{{torque3, 0}, None},
{{tick, False}, None},
{{vEnd, {0, 0, 0}}, None},
TrackedSymbols :→ {tick},
SynchronousUpdating → False,
(*DisplayAllSteps→True,*)
ControlPlacement → Left,
Initialization :→
 (
  integerStrictPositive = (IntegerQ[#] && # > 0 &);
  integerPositive = (IntegerQ[#] && # ≥ 0 &);
  numericStrictPositive = (Element[#, Reals] && # > 0 &);
  numericPositive = (Element[#, Reals] && # ≥ 0 &);
  numericStrictNegative = (Element[#, Reals] && # < 0 &);
  numericNegative = (Element[#, Reals] && # ≤ 0 &);
  bool = (Element[#, Booleans] &);
```

```
numeric = (Element[#, Reals] &);
integer = (Element[#, Integers] &);
padIt1[v_?numeric, f_List] := AccountingForm[v, f,
  NumberSigns → {"-", "+"}, NumberPadding → {"0", "0"}, SignPadding → True];
padIt1[v_?numeric, f_Integer] := AccountingForm[Chop[v], f,
  NumberSigns → {"-", "+"}, NumberPadding → {"0", "0"}, SignPadding → True];
padIt2[v_?numeric, f_List] := AccountingForm[v, f, NumberSigns → {"", ""},
  NumberPadding → {"0", "0"}, SignPadding → True];
padIt2[v_?numeric, f_Integer] := AccountingForm[Chop[v], f,
  NumberSigns → {"", ""}, NumberPadding → {"0", "0"}, SignPadding → True];

zDot[{q1_, q2_, q3_, qd1_, qd2_, qd3_}, c1_, c2_, c3_, m1_, m2_,
  m3_, L1_, L2_, L3_, r1_, r2_, r3_, torque1_, torque2_, torque3_] :=
 Module[{D0, B0, C0, G0, friction, qdd},
  D0 = getMassMatrix[q1, q2, q3, c1, c2, c3, m1, m2, m3, L1, L2, L3, r1, r2, r3];
  B0 = getB[q1, q2, q3, c1, c2, c3, m1, m2, m3, L1, L2, L3, r1, r2, r3];
  C0 = getC[q1, q2, q3, c1, c2, c3, m1, m2, m3, L1, L2, L3, r1, r2, r3];
  G0 = getG[q1, q2, q3, c1, c2, c3, m1, m2, m3, L1, L2, L3, r1, r2, r3];
  friction = {c1 * qd1, c2 * qd2, c3 * qd3};
  qdd = Inverse[D0].(-B0.{qd1 * qd2, qd1 * qd3, qd2 * qd3} -
      C0.{qd1^2, qd2^2, qd3^2} - G0 - friction + {torque1, torque2, torque3});
  Flatten@{qd1, qd2, qd3, qdd}
 ];

getMassMatrix[q1_, q2_, q3_, c1_,
  c2_, c3_, m1_, m2_, m3_, L1_, L2_, L3_, r1_, r2_, r3_] :=
 {{(1/24) * (4 * L2^2 * m2 + 12 * L2^2 * m3 + 4 * L3^2 * m3 + 12 * m1 * r1^2 +
      9 * m2 * r2^2 + 9 * m3 * r3^2 + (4 * L2^2 * (m2 + 3 * m3) - 3 * m2 * r2^2) *
       Cos[2 * q2] + 12 * L2 * L3 * m3 * Cos[q3] +
 4 * L3^2 * m3 * Cos[2 * (q2 + q3)] - 3 * m3 * r3^2 * Cos[2 * (q2 + q3)] +
      12 * L2 * L3 * m3 * Cos[2 * q2 + q3]), 0, 0},
{0, (1/12) * (4 * L3^2 * m3 + 4 * L2^2 * (m2 + 3 * m3) + 3 * m2 * r2^2 + 3 * m3 * r3^2) +
    L2 * L3 * m3 * Cos[q3], (1/12) * m3 * (4 * L3^2 + 3 * r3^2 + 6 * L2 * L3 * Cos[q3])},
{0, (1/12) * m3 * (4 * L3^2 + 3 * r3^2 + 6 * L2 * L3 * Cos[q3]),
   (1/12) * m3 * (4 * L3^2 + 3 * r3^2)}};

getB[q1_, q2_, q3_, c1_, c2_, c3_, m1_, m2_, m3_, L1_, L2_, L3_, r1_, r2_, r3_] :=
 {{(1/12) * ((-4 * L2^2 * (m2 + 3 * m3) + 3 * m2 * r2^2) * Sin[2 * q2] + m3 *
      ((-4 * L3^2 + 3 * r3^2) * Sin[2 * (q2 + q3)] - 12 * L2 * L3 * Sin[2 * q2 + q3])),
(-(1/6)) * m3 * (6 * L2 * L3 * Cos[q2] + (4 * L3^2 - 3 * r3^2) * Cos[q2 + q3]) *
    Sin[q2 + q3], 0}, {0, 0, (-L2) * L3 * m3 * Sin[q3]}, {0, 0, 0}};

getC[q1_, q2_, q3_, c1_, c2_, c3_, m1_, m2_, m3_, L1_, L2_, L3_, r1_, r2_, r3_] :=
```

```
 {{0, 0, 0},
  {(1/24) * ((4 * L2^2 * (m2 + 3 * m3) - 3 * m2 * r2^2) * Sin[2 * q2] + m3 * ((4 * L3^2 -
            3 * r3^2) * Sin[2 * (q2 + q3)] + 12 * L2 * L3 * Sin[2 * q2 + q3])), 0,
(-(1/2)) * L2 * L3 * m3 * Sin[q3]}, {(1/12) * m3 * (6 * L2 * L3 * Cos[q2] +
       (4 * L3^2 - 3 * r3^2) * Cos[q2 + q3]) *
     Sin[q2 + q3], (1/2) * L2 * L3 * m3 * Sin[q3], 0}};

getG[q1_, q2_, q3_, c1_, c2_, c3_, m1_, m2_, m3_, L1_, L2_, L3_, r1_, r2_, r3_] :=
 {0, (1/2) * 9.8 * (L2 * (m2 + 2 * m3) * Cos[q2] + L3 * m3 * Cos[q2 + q3]),
  (1/2) * 9.8 * L3 * m3 * Cos[q2 + q3]};


getV3[L1_, L2_, L3_, x1_, x2_, x3_, v1_, v2_, v3_] :=
 {-(L2 Cos[x2] + L3 Cos[x2 + x3]) Sin[x1] v1 -
   Cos[x1] ((L2 Sin[x2] + L3 Sin[x2 + x3]) v2 + L3 Sin[x2 + x3] v3),
  Cos[x1] (L2 Cos[x2] + L3 Cos[x2 + x3]) v1 -
   Sin[x1] ((L2 Sin[x2] + L3 Sin[x2 + x3]) v2 + L3 Sin[x2 + x3] v3),
  (L2 Cos[x2] + L3 Cos[x2 + x3]) v2 + L3 Cos[x2 + x3] v3};

o02[x1_, x2_, x3_, L1_, L2_, L3_] :=
 {L2 Cos[x1] Cos[x2], L2 Cos[x2] Sin[x1], L1 + L2 Sin[x2]};
o03[x1_, x2_, x3_, L1_, L2_, L3_] :=
 {L2 Cos[x1] Cos[x2] + L3 Cos[x1] Cos[x2] Cos[x3] - L3 Cos[x1] Sin[x2] Sin[x3],
  L2 Cos[x2] Sin[x1] + L3 Cos[x2] Cos[x3] Sin[x1] - L3 Sin[x1] Sin[x2] Sin[x3],
  L1 + L2 Sin[x2] + L3 Cos[x3] Sin[x2] + L3 Cos[x2] Sin[x3]};

typeOne[r_, {x_, y_, z_}, L0_, flag_, upper_] := Module[{},
  (*flag says to add a cylinder at bottom,
  upper is a flag to say to add a cylinder at top*)
  {Cuboid[{x - r/2, y - r/2, z}, {x + r/2, y + r/2, z + L0}],
   If[upper,
    (*left and right cylinders*)
    {
     Cylinder[{{x - 0.5 r, y, z + L0 + 0.3 r}, {x - 0.4 r, y, z + L0 + 0.3 r}}, .6 r],
     Cylinder[{{x + 0.4 r, y, z + L0 + 0.3 r}, {x + 0.5 r, y, z + L0 + 0.3 r}}, .6 r],
     (*inner one*)
     Cylinder[{{x - .8 r, y, z + L0 + 0.3 r}, {x + 0.8 r, y, L0 + z + .3 r}}, .1 r]
    }
    ,
    {
     Cuboid[{x - 0.5, y - r/2, z + L0}, {x - 0.4 r, y + r/2, z + L0 + r}],
     Cuboid[{x + 0.4, y - r/2, z + L0}, {x + 0.5 r, y + r/2, z + L0 + r}]
    }
```

```
          ],
          If[flag, Cylinder[{{x - .6 r, y, z}, {x + 0.6 r, y, z}}, .5 r], ## &[]]
         }
        ];

   )
]
```