Mahdieh Nejati Javaremi

ME 454 : Numerical Methods in Optimal Control

Final Project _ Spring 2015

```
Quit[];
ClearSystemCache[];
```

Double Pendulum

Deriving the Equations of Motion

```
x1[t] = R1 Sin[q1[t]];
y1[t] = -R1 Cos[q1[t]];
x2[t] = R1 Sin[q1[t]] + R2 Sin[q1[t] + q2[t]];
y2[t] = -R1 Cos[q1[t]] - R2 Cos[q1[t] + q2[t]];
vx1 = D[x1[t], t];
vy1 = D[y1[t], t];
vx2 = D[x2[t], t];
vy2 = D[y2[t], t];
Lag = FullSimplify[
   1/2 m1 (vx1^2 + vy1^2) + 1/2 m2 (vx2^2 + vy2^2) - m1 g y1[t] - m2 g y2[t]];
EQ1 = FullSimplify[D[D[Lag, q1'[t]], t] - D[Lag, q1[t]]];
EQ2 = FullSimplify[D[D[Lag, q2'[t]], t] - D[Lag, q2[t]]];
EQ = Solve[{EQ1 == u1[t], EQ2 == u2[t]}, {q1''[t], q2''[t]}] // FullSimplify;
```

Solving the Optimization

```
g = 9.81;
h = 1;
```

```
T = 2;
R1 = 0.5;
R2 = 0.5;
m1 = 0.5;
m2 = 0.5;

(* State and Control *)
X = {{θ1[t]}, {θ2[t]}, {θ1'[t]}, {θ2'[t]}};
dX = D[X, {t, 1}];
U = {{u1[t]}, {u2[t]}};
(* Desired Trajectories *)
θ1d[t_] := Sin[π t];
θ2d[t_] := Sin[π t];
dθ1d[t_] := π Cos[π t];
dθ2d[t_] := π Cos[π t];
Xd[t_] := {{θ1d[t]}, {θ2d[t]}, {dθ1d[t]}, {dθ2d[t]}};

Q = 10 * IdentityMatrix[4];
Qn = Q;
Qr = Q;

R = 0.001 * IdentityMatrix[2];
Rn = R;
Rr = R;

P1 = 0 * IdentityMatrix[4];
P1n = P1;
P1r = P1;

L[X_, U_] := 1/2 ((X - Xd[t])ᵀ.Q.(X - Xd[t])) + 1/2 Uᵀ.R.U;
J[X_, U_] := Quiet[NIntegrate[L[X, U], {t, 0, T},
      Method → {Automatic, "SymbolicProcessing" → False}]] +
   1/2 ((X - Xd[t]) /. t → T)ᵀ.P1.((X - Xd[t]) /. t → T);
f[x_, u_] := {{x[[3, 1]]}, {x[[4, 1]]},
   {(2 R2 u[[1, 1]] - 2 (R2 + R1 Cos[x[[2, 1]]]) u[[2, 1]] +
        R1 R2 (-g (2 m1 + m2) Sin[x[[1, 1]]] + g m2 Sin[x[[1, 1]] + 2 x[[2, 1]]] +
            m2 R1 Sin[2 x[[2, 1]]] x[[3, 1]]^2 + 2 m2 R2 Sin[x[[2, 1]]]
             (x[[3, 1]] + x[[4, 1]])^2)) / (2 R1^2 R2 (m1 + m2 - m2 Cos[x[[2, 1]]]^2))},
    {(g m2 R1 R2 (-R2 (2 m1 + m2 - m2 Cos[2 x[[2, 1]]]) Sin[x[[1, 1]]] +
            2 Cos[x[[1, 1]]] ((m1 + m2) R1 + m2 R2 Cos[x[[2, 1]]]) Sin[x[[2, 1]]]) +
        2 m2 R2 (R2 + R1 Cos[x[[2, 1]]]) u[[1, 1]] +
        2 ((m1 + m2) R1^2 + m2 R2^2 + 2 m2 R1 R2 Cos[x[[2, 1]]])
```

```
        (-u[[2, 1]] + m2 R1 R2 Sin[x[[2, 1]]] x[[3, 1]]^2) +
       4 m2^2 R1 R2^2 (R2 + R1 Cos[x[[2, 1]]]) Sin[x[[2, 1]]] x[[4, 1]] x[[3, 1]] +
       2 m2^2 R1 R2^2 (R2 + R1 Cos[x[[2, 1]]]) Sin[x[[2, 1]]] x[[4, 1]]^2) /
     (m2 R1^2 R2^2 (-2 m1 - m2 + m2 Cos[2 x[[2, 1]]]))}};
DJzeta[xi_, zeta_] := Module[{X = xi[[1]], U = xi[[2]],
    z = zeta[[1]], v = zeta[[2]]},
   Return[Quiet[NIntegrate[(Q.(X - Xd[t]))^T.z + (R.U)^T.v, {t, 0, T},
        Method → {Automatic, "SymbolicProcessing" → False}]] +
      ((P1.(X - Xd[t]))^T.z) /. t → T];
  ];

xibar0 = {Xd[t], {{0}, {0}}};
xi0 = {{{1}, {1}, {0}, {0}}, {{0}, {0}}};

Asym = D[(f[X, U]), X^T];
Bsym = D[f[X, U], U^T];
asym = D[L[X, U], {X, 1}][[1, 1]];
bsym = D[L[X, U], {U, 1}][[1, 1]];
Ta[R_] := Table[R[[i, 1]], {i, 1, 4}];

(* Riccati solution for P *)
Psol[A_, B_, Q_ , R_, P1_] := Module[{PEQ1, PEQ2, Ps, i, j},
   Ri[t_] := Table[P_{i,j}[t], {i, 1, 4}, {j, 1, 4}];
   PEQ1 = (Ri'[t] + A^T.Ri[t] + Ri[t].A - Ri[t].B.Inverse[R].B^T.Ri[t] + Q) ==
     {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}};
   PEQ2 = Ri[T] == P1;
   Ps = (NDSolve[{PEQ1, PEQ2}, Flatten[Ri[t]], {t, 0, T}])[[1]];
   Return[Ri[t] /. Ps];
  ];

(* Riccati solution for r *)
rsol[A_, B_, a_, b_, P_, R_, P1_, xi_] := Module[{rEQ1, rEQ2, rs},
   Rir[t_] := {{r1[t]}, {r2[t]}, {r3[t]}, {r4[t]}};
   rEQ1 =
    Rir'[t] + (A - B.Inverse[R].B^T.P)^T.Rir[t] + a - P.B.Inverse[R].b ==
     {{0}, {0}, {0}, {0}};
   rEQ2 = Rir[T] == (P1.(xi[[1]] - Xd[t])) /. t → T;
   rs = (NDSolve[{rEQ1, rEQ2}, Flatten[Rir[t]], {t, 0, T}])[[1]];
   Return[Rir[t] /. rs];
  ];

(* Descent Direction solution *)
zsol[A_, B_, b_, P_, r_] := Module[{v, zEQ1, zEQ2, zs},
```

```
    v = -Inverse[Rn].(b + BT.P.{{z1[t]}, {z2[t]}, {z3[t]}, {z4[t]}} + BT.r);
    zEQ1 = {{z1'[t]}, {z2'[t]}, {z3'[t]}, {z4'[t]}} ==
      A.{{z1[t]}, {z2[t]}, {z3[t]}, {z4[t]}} + B.v;
    zEQ2 = {{z1[0]}, {z2[0]}, {z3[0]}, {z4[0]}} == {{0}, {0}, {0}, {0}};
    (* zEQ2 = {{z1[0]},{z2[0]}}== -Inverse[P... *)
    zs = (NDSolve[{zEQ1, zEQ2}, {z1[t], z2[t], {z3[t]}, {z4[t]}},
         {t, 0, T}])[[1]];
    Return[({{z1[t]}, {z2[t]}, {z3[t]}, {z4[t]}} /. zs)];
   ];

(* Projection of xibar onto feasible space *)
Proj[xibar_, K_] :=
  Module[{xbar = xibar[[1]], ubar = xibar[[2]], xEQ1, xEQ2, xs},
   xEQ1 = {dX[[3]], dX[[4]]} == {(f[X, U] /. {u1[t] → (ubar + K.(X - xbar))[[1, 1]],
           u2[t] → (ubar + K.(X - xbar))[[2, 1]]})[[3]],
       (f[X, U] /. {u1[t] → (ubar + K.(X - xbar))[[1, 1]],
           u2[t] → (ubar + K.(X - xbar))[[2, 1]]})[[4]]};
   xEQ2 = {{θ1[0]}, {θ2[0]}, {θ1'[0]}, {θ2'[0]}} == {{0}, {0}, {0}, {0}};
   xs =
    (NDSolve[{xEQ1, xEQ2}, {θ1[t], θ2[t], θ1'[t], θ2'[t]}, {t, 0, T}])[[1]];
   Return[(X /. xs)];
  ];

(* Combining interpolationg functions by sampling, significanlty
 reduces computation time as the number of iterations increases *)
combineInterps[interp_, maxIndex_, stepSize_] :=
  Module[{samples, index, val, retInterp}, samples = {};
   index = 0;
   While[index ≤ maxIndex, AppendTo[samples, {index, interp /. t → index}];
    index += stepSize;];
   Return[{Interpolation[samples, Method → "Hermite"][t]}];];

(* Armijo Line Search *)
Armijo[xi_, xibar_, zeta_, K_, maxIters_: 20] :=
  Module[{α = .0001, β = .7, n = 0, xibarn, γ, X, U, Xn, Un, Jtemp, DJtemp},
   X = xi[[1]];
   U = xi[[2]];
   γ = β^n;
   xibarn = xibar + γ zeta;
   Xn  = Proj[xibarn, K];
   Un = xibarn[[2]] + K.(Xn - xibarn[[1]]);
   Un = {combineInterps[Un[[1, 1]], T, 1/100],
     combineInterps[Un[[2, 1]], T, 1/100]};
```

```
    Jtemp = J[X, U]; (* Only changes in main loop *)
    DJtemp = DJzeta[xi, zeta]; (* Only changes in main loop *)
    While[
     And[(J[Xn, Un])[[1, 1]] > (Jtemp + α γ DJtemp )[[1, 1]], n < maxIters],
     n = n + 1;
     γ = β^n;
     xibarn = xibar + γ zeta;
     Xn = Proj[xibarn, K];
     Un = xibarn[[2]] + K.(Xn - xibarn[[1]]);
     Un = {combineInterps[Un[[1, 1]], T, 1/100],
       combineInterps[Un[[2, 1]], T, 1/100]};
     Print["γ: ", γ];
    ];
    Return[{xibarn, {Xn, Un}}];
   ];

ϵ = 0.6684;
i = 0;
normᵢ = 100;

(* Full Algorithm *)
While[
  And[Abs[normᵢ] > ϵ, i < 30],
  A = Ta[Asym] /.
    {θ1[t] → xiᵢ[[1, 1, 1]], θ2[t] → xiᵢ[[1, 2, 1]], θ1'[t] → xiᵢ[[1, 3, 1]],
     θ2'[t] → xiᵢ[[1, 4, 1]], u1[t] → xiᵢ[[2, 1, 1]], u2[t] → xiᵢ[[2, 2, 1]]};
  A = {Flatten[{combineInterps[A[[1, 1]], T, 1/100],
      combineInterps[A[[1, 2]], T, 1/100], combineInterps[A[[1, 3]], T, 1/100],
      combineInterps[A[[1, 4]], T, 1/100]}],
    Flatten[{combineInterps[A[[2, 1]], T, 1/100],
      combineInterps[A[[2, 2]], T, 1/100], combineInterps[A[[2, 3]], T, 1/100],
      combineInterps[A[[2, 4]], T, 1/100]}],
    Flatten[{combineInterps[A[[3, 1]], T, 1/100],
      combineInterps[A[[3, 2]], T, 1/100], combineInterps[A[[3, 3]], T, 1/100],
      combineInterps[A[[3, 4]], T, 1/100]}],
    Flatten[{combineInterps[A[[4, 1]], T, 1/100],
      combineInterps[A[[4, 2]], T, 1/100], combineInterps[A[[4, 3]], T, 1/100],
      combineInterps[A[[4, 4]], T, 1/100]}]};
  B = Ta[Bsym] /. {θ2[t] → xiᵢ[[1, 2, 1]]};
  B = {Flatten[{combineInterps[B[[1, 1]], T, 1/100],
      combineInterps[B[[1, 2]], T, 1/100]}],
    Flatten[{combineInterps[B[[2, 1]], T, 1/100], combineInterps[
       B[[2, 2]], T, 1/100]}], Flatten[{combineInterps[B[[3, 1]], T, 1/100],
```

```
      combineInterps[B[[3, 2]], T, 1/100]}], Flatten[{combineInterps[
       B[[4, 1]], T, 1/100], combineInterps[B[[4, 2]], T, 1/100]}]};
  a = asym /. {θ1[t] → xii[[1, 1, 1]], θ2[t] → xii[[1, 2, 1]],
     θ1'[t] → xii[[1, 3, 1]], θ2'[t] → xii[[1, 4, 1]]};
  b = bsym /. {u1[t] → xii[[2, 1, 1]], u2[t] → xii[[2, 2, 1]]};
  Pni = Psol[A, B, Qn, Rn, P1n];
  ri = rsol[A, B, a, b, Pni, Rn, P1n, xii];
  zi = zsol[A, B, b, Pni, ri];
  vi = -Inverse[Rn].(b + B^T.Pni.zi + B^T.ri);
  vi = {combineInterps[vi[[1, 1]], T, 1/100],
    combineInterps[vi[[2, 1]], T, 1/100]};
  zetai = {zi, vi};
  Pri = Psol[A, B, Qr, Rr, P1r];
  κi = -Inverse[Rr].B^T.Pri;
  κi = {Flatten[{combineInterps[κi[[1, 1]], T, 1/100],
      combineInterps[κi[[1, 2]], T, 1/100], combineInterps[
       κi[[1, 3]], T, 1/100], combineInterps[κi[[1, 4]], T, 1/100]}],
    Flatten[{combineInterps[κi[[2, 1]], T, 1/100], combineInterps[
       κi[[2, 2]], T, 1/100], combineInterps[κi[[2, 3]], T, 1/100],
      combineInterps[κi[[2, 4]], T, 1/100]}]};
  {xibari+1, xii+1} = Armijo[xii, xibari, zetai, κi];
  xibari+1 = {{combineInterps[xibari+1[[1, 1, 1]], T, 1/100],
     combineInterps[xibari+1[[1, 2, 1]], T, 1/100],
     combineInterps[xibari+1[[1, 3, 1]], T, 1/100],
     combineInterps[xibari+1[[1, 4, 1]], T, 1/100]},
    {combineInterps[xibari+1[[2, 1, 1]], T, 1/100],
     combineInterps[xibari+1[[2, 2, 1]], T, 1/100]}};
  xii+1 = {{combineInterps[xii+1[[1, 1, 1]], T, 1/100],
     combineInterps[xii+1[[1, 2, 1]], T, 1/100], combineInterps[
      xii+1[[1, 3, 1]], T, 1/100], combineInterps[xii+1[[1, 4, 1]], T, 1/100]},
    {combineInterps[xii+1[[2, 1, 1]], T, 1/100],
     combineInterps[xii+1[[2, 2, 1]], T, 1/100]}};
  normi+1 = (DJzeta[xii, zetai])[[1, 1]];
  (*Print[xii+1[[1]]/.t→ T];*)
  Print["norm: ", normi+1];
  i = i + 1;
 ];

Print["Number of iterations: ", i];
(* Plot the trajectories and control effort *)
Plot[{θ1d[t], xii[[1, 1, 1]]}, {t, 0, T}, PlotRange → Full ,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"θ1 desired", "θ1 actaul"}]
Plot[{θ2d[t], xii[[1, 2, 1]]}, {t, 0, T}, PlotRange → Full ,
```

```
 PlotLabel → "Desired vs. Actual", PlotLegends → {"θ2 desired", "θ2 actaul"}]
Plot[{dθ1d[t], xiᵢ[[1, 3, 1]]}, {t, 0, T},
 PlotRange → Full , PlotLabel → "Desired vs. Actual",
 PlotLegends → {"θ1' desired", "θ1' actaul"}]
Plot[{dθ2d[t], xiᵢ[[1, 4, 1]]}, {t, 0, T},
 PlotRange → Full , PlotLabel → "Desired vs. Actual",
 PlotLegends → {"θ2' desired", "θ2' actaul"}]
Plot[{u1d[t], xiᵢ[[2, 1, 1]]}, {t, 0, T}, PlotRange → Full,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"u1 desired ", "u1 actual"}]
Plot[{u2d[t], xiᵢ[[2, 2, 1]]}, {t, 0, T}, PlotRange → Full,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"u2 desired ", "u2 actual"}]
ListLinePlot[Table[{h, Abs[normₕ]}, {h, 1, i}], Filling → Axis]

(*ANIMATION*)
(*x and y coordinates for pendulum 1 *)
R1 = R2 = 1;
X1[τ_] := R1 * Sin[xiᵢ[[1, 1, 1]]] /. t → τ;
Y1[τ_] := -R1 * Cos[xiᵢ[[1, 1, 1]]] /. t → τ;
(*x and y coordinates for pendulum 2*)
X2[τ_] :=
  R1 * Sin[xiᵢ[[1, 1, 1]]] + R2 * Sin[xiᵢ[[1, 1, 1]] + xiᵢ[[1, 2, 1]]] /. t → τ;
Y2[τ_] := -R1 * Cos[xiᵢ[[1, 1, 1]]] -
    R2 * Cos[xiᵢ[[1, 1, 1]] + xiᵢ[[1, 2, 1]]] /. t → τ;

Animate[Show [Graphics[{PointSize[0.06], Orange, Point[{ X1[τ], Y1[τ]}], Pink,
    Point[{X2[τ], Y2[τ]}], Purple, Thick, Line[{{0, 0}, {X1[τ], Y1[τ]}}],
    Purple, Thick, Line[{{X1[τ], Y1[τ]}, {X2[τ], Y2[τ]}}],
    Black, Line[{{-3, 0}, {3, 0}}], Line[{{-3, 0}, {-3, -2}}]}],
  AspectRatio → Automatic, PlotRange → {{-2.1, 2.1}, {-2.1, 2.1}},
  Frame → True], {τ, 0, 2}, AnimationRate → 1]

(*The trajectory plot of the Double Pendulum*)
ParametricPlot[{{X1[τ], Y1[τ]}, {X2[τ], Y2[τ]}},
 {τ, 0, 2}, AspectRatio → Automatic, AxesLabel → {x, y},
 PlotLegends → {"Trajectory of pendulum 1", "Trajectory of pendulum 2"}]
```

RR Arm

Deriving the Equations of Motion

```
(* fully actuated *)
q = {{θ1[t]}, {θ2[t]}};
dq = D[q, t];
ddq = D[dq, t];
X = {{θ1[t]}, {θ2[t]}, {θ1'[t]}, {θ2'[t]}};
U = {{u1[t]}, {u2[t]}};
(* mass  matrix *)
MM[x_] :=
  {{α + 2 β Cos[x[[2, 1]]], δ + β Cos[x[[2, 1]]]}, {δ + β Cos[x[[2, 1]]], δ}};
(* vector covering corriolis effects *)
CC[x_] :=
  {{-β Sin[x[[2, 1]]] x[[4, 1]] + b1, -β Sin[x[[2, 1]]] (x[[3, 1]] + x[[4, 1]])},
   {β Sin[x[[2, 1]]] x[[3, 1]], b2}};
(* vector of gravitational forces *)
GG[x_] := {{m1 g r1 Cos[x[[1, 1]]] +
      m2 g (l1 Cos[x[[1, 1]]] + r2 Cos[x[[1, 1]] + x[[2, 1]]])},
   {m2 g r2 Cos[x[[1, 1]] + x[[2, 1]]]}};
(* vector of input torques *)
tau = {{τ1}, {τ2}};
EQ[x_, u_] :=
  Solve[{MM[x].(D[D[x, t], t][[1 ;; 2]] - u) + CC[x].dq + GG[x] == tau},
    {D[D[x, t], t][[1, 1]], D[D[x, t], t][[2, 1]]}] // FullSimplify;
EQ[X, U][[1, 1]];
EQ[X, U][[1, 2]];
```

Solving the Optimization

In[1]:=

```
g = 9.81;
T = 5;

(* link lenghts *)
ll1 = 0.5; ll2 = 0.5;
(* mass of each link *)
mm1 = 0.5; mm2 = 0.5;
(* Distance to center of mass for each link *)
rr1 = 0.25; rr2 = 0.25;
(* Damping coefficient of joints *)
bb1 = 0 ; bb2 = 0;
(* moments of inertia about z-axis *)
Iz1 = 0.01; Iz2 = 0.01;
```

```
αα = Iz1 + Iz2 + mm1 rr1^2 + mm2 (ll1^2 + ll2^2);
ββ = mm2 ll2 rr2;
δδ = Iz2 + mm2 rr2^2;

(* State and Control *)
X = {{θ1[t]}, {θ2[t]}, {θ1'[t]}, {θ2'[t]}};
dX = D[X, {t, 1}];
U = {{u1[t]}, {u2[t]}};

(** Desired Trajectory **)
θ1d[t_] := Sin[π t];
θ2d[t_] := Sin[π t];
dθ1d[t_] := π Cos[π t];
dθ2d[t_] := π Cos[π t];

(** Desired Trajectory **)
(*
(* Piecewise trajectory *)
αd = 45*Pi/180;
θ1d[t_]:= Piecewise[
  {{1, 0<t<1},{1+(αd/2)*(1 - Cos[(2 Pi/4)*(t-3)]),1≤t<3},{1, 3≤t<5}}];
θ2d[t_]:= Piecewise[{{1, 0<t<1},
   {1+(αd/2)*(1 - Cos[(2 Pi/4)*(t-3)]),1≤t<3},{1, 3≤t<5}}];
dθ1d[t_]:=Piecewise[{{0, 0<t<1},{π*(αd/4)*Sin[π*(t-3)],1≤t<3},{0, 3≤t<5}}];
dθ2d[t_]:=Piecewise[{{0, 0<t<1},{π*(αd/4)*Sin[π*(t-3)],1≤t<3},{0, 3≤t<5}}];
*)

Xd[t_] := {{θ1d[t]}, {θ2d[t]}, {dθ1d[t]}, {dθ2d[t]}};
Ud[t_] := {{0}, {0}};

Q = 10 * IdentityMatrix[4];
Qn = Q;
Qr = Q;

R = 0.001 * IdentityMatrix[2];
Rn = R;
Rr = R;

P1 = 0 * IdentityMatrix[4];
P1n = P1;
P1r = P1;
```

```
L[X_, U_] := 1/2 ((X - Xd[t])^T.Q.(X - Xd[t])) + 1/2 U^T.R.U ;
J[X_, U_] := Quiet[NIntegrate[L[X, U], {t, 0, T},
      Method → {Automatic, "SymbolicProcessing" → False}]] +
    1/2 ((X - Xd[t]) /. t → T)^T.P1.((X - Xd[t]) /. t → T);
f[x_, u_] := {
    {x[[3, 1]]},
    {x[[4, 1]]}, {1/((αα - δδ) δδ - ββ^2 Cos[x[[2, 1]]]^2)
      (δδ (u[[1, 1]] - u[[2, 1]]) - g (ll1 mm2 + mm1 rr1) δδ Cos[x[[1, 1]]] +
        ββ Cos[x[[2, 1]]] (-u[[2, 1]] + g mm2 rr2 Cos[x[[1, 1]] + x[[2, 1]]]) +
        ββ^2 Cos[x[[2, 1]]] Sin[x[[2, 1]]] x[[3, 1]]^2 + bb2 ββ Cos[x[[2, 1]]]
         x[[4, 1]] + ββ δδ Sin[x[[2, 1]]] (x[[3, 1]] + x[[4, 1]])^2 +
        δδ (-bb1 x[[3, 1]] + bb2 x[[4, 1]]))}, {-1/((αα - δδ) δδ - ββ^2 Cos[x[[2, 1]]]^2)
      (δδ u[[1, 1]] - αα u[[2, 1]] + ββ (u[[1, 1]] - 2 u[[2, 1]]) Cos[x[[2, 1]]] -
        g (ll1 mm2 + mm1 rr1) Cos[x[[1, 1]]] (δδ + ββ Cos[x[[2, 1]]]) +
        g mm2 rr2 (αα - δδ + ββ Cos[x[[2, 1]]]) Cos[x[[1, 1]] + x[[2, 1]]] +
        ββ (αα + 2 ββ Cos[x[[2, 1]]]) Sin[x[[2, 1]]] x[[3, 1]]^2 +
        bb2 (αα + 2 ββ Cos[x[[2, 1]]]) x[[4, 1]] +
        ββ (δδ + ββ Cos[x[[2, 1]]]) Sin[x[[2, 1]]] x[[4, 1]]^2 +
        (δδ + ββ Cos[x[[2, 1]]]) x[[3, 1]] (-bb1 + 2 ββ Sin[x[[2, 1]]] x[[4, 1]]))}
  };
DJzeta[xi_, zeta_] :=
  Module[{X = xi[[1]], U = xi[[2]], z = zeta[[1]], v = zeta[[2]]},
   Return[Quiet[NIntegrate[(Q.(X - Xd[t]))^T.z + (R.U)^T.v, {t, 0, T},
         Method → {Automatic, "SymbolicProcessing" → False}]] +
       ((P1.(X - Xd[t]))^T.z) /. t → T];
  ];

xibar0 = {Xd[t], Ud[t]};
xi0 = {{{1}, {1}, {0}, {0}}, {{0}, {0}}};

(* Symbolic forms of A, B, a, b. Ta transforms A and B to the proper size *)
Asym = D[(f[X, U]), X^T];
Bsym = D[f[X, U], U^T];
asym = D[L[X, U], {X, 1}][[1, 1]];
bsym = D[L[X, U], {U, 1}][[1, 1]];
Ta[R_] := Table[R[[i, 1]], {i, 1, 4}];

(* Riccati solution for P *)
Psol[A_, B_, Q_ , R_, P1_] := Module[{PEQ1, PEQ2, Ps, i, j},
   Ri[t_] := Table[P_{i,j}[t], {i, 1, 4}, {j, 1, 4}];
   PEQ1 = (Ri'[t] + A^T.Ri[t] + Ri[t].A - Ri[t].B.Inverse[R].B^T.Ri[t] + Q) ==
```

```
      {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}};
    PEQ2 = Ri[T] == P1;
    Ps = (NDSolve[{PEQ1, PEQ2}, Flatten[Ri[t]], {t, 0, T}])[[1]];
    Return[Ri[t] /. Ps];
   ];

(* Riccati solution for r *)
rsol[A_, B_, a_, b_, P_, R_, P1_, xi_] := Module[{rEQ1, rEQ2, rs},
    Rir[t_] := {{r1[t]}, {r2[t]}, {r3[t]}, {r4[t]}};
    rEQ1 =
     Rir'[t] + (A - B.Inverse[R].Bᵀ.P)ᵀ.Rir[t] + a - P.B.Inverse[R].b ==
      {{0}, {0}, {0}, {0}};
    rEQ2 = Rir[T] == (P1.(xi[[1]] - Xd[t])) /. t → T;
    rs = (NDSolve[{rEQ1, rEQ2}, Flatten[Rir[t]], {t, 0, T}])[[1]];
    Return[Rir[t] /. rs];
   ];

(* Descent Direction solution *)
zsol[A_, B_, b_, P_, r_] := Module[{v, zEQ1, zEQ2, zs},
    v = -Inverse[Rn].(b + Bᵀ.P.{{z1[t]}, {z2[t]}, {z3[t]}, {z4[t]}} + Bᵀ.r);
    zEQ1 = {{z1'[t]}, {z2'[t]}, {z3'[t]}, {z4'[t]}} ==
      A.{{z1[t]}, {z2[t]}, {z3[t]}, {z4[t]}} + B.v;
    zEQ2 = {{z1[0]}, {z2[0]}, {z3[0]}, {z4[0]}} == {{0}, {0}, {0}, {0}};
    (* zEQ2 = {{z1[0]},{z2[0]}}== -Inverse[P... *)
    zs = (NDSolve[{zEQ1, zEQ2}, {z1[t], z2[t], {z3[t]}, {z4[t]}},
         {t, 0, T}])[[1]];
    Return[({{z1[t]}, {z2[t]}, {z3[t]}, {z4[t]}} /. zs)];
   ];

(* Projection of xibar onto feasible space *)
Proj[xibar_, K_] :=
  Module[{xbar = xibar[[1]], ubar = xibar[[2]], xEQ1, xEQ2, xs},
    xEQ1 = {dX[[3]], dX[[4]]} == {(f[X, U] /. {u1[t] → (ubar + K.(X - xbar))[[1, 1]],
           u2[t] → (ubar + K.(X - xbar))[[2, 1]]})[[3]],
       (f[X, U] /. {u1[t] → (ubar + K.(X - xbar))[[1, 1]],
           u2[t] → (ubar + K.(X - xbar))[[2, 1]]})[[4]]};
    xEQ2 = {{θ1[0]}, {θ2[0]}, {θ1'[0]}, {θ2'[0]}} == xi0[[1]];
    xs =
     (NDSolve[{xEQ1, xEQ2}, {θ1[t], θ2[t], θ1'[t], θ2'[t]}, {t, 0, T}])[[1]];
    Return[(X /. xs)];
   ];

combineInterps[interp_, maxIndex_, stepSize_] :=
```

```
  Module[{samples, index, val, retInterp}, samples = {};
   index = 0;
   While[index ≤ maxIndex, AppendTo[samples, {index, interp /. t → index}];
    index += stepSize;];
   Return[{Interpolation[samples, Method → "Hermite"][t]}];];

(* Armijo Line Search *)
Armijo[xi_, xibar_, zeta_, K_, maxIters_: 10] :=
  Module[{α = .01, β = .5, n = 0, xibarn, γ, X, U, Xn, Un, Jtemp, DJtemp},
   X = xi[[1]];
   U = xi[[2]];
   γ = β^n;
   xibarn = xibar + γ zeta;
   Xn  = Proj[xibarn, K];
   Un  = xibarn[[2]] + K.(Xn - xibarn[[1]]);
   Un = {combineInterps[Un[[1, 1]], T, 1/100],
     combineInterps[Un[[2, 1]], T, 1/100]};
   Jtemp = J[X, U]; (* Only changes in main loop *)
   DJtemp = DJzeta[xi, zeta]; (* Only changes in main loop *)
   While[
    And[(J[Xn, Un])[[1, 1]] > (Jtemp + α γ DJtemp )[[1, 1]], n < maxIters],
    n = n + 1;
    γ = β^n;
    xibarn = xibar + γ zeta;
    Xn  = Proj[xibarn, K];
    Un  = xibarn[[2]] + K.(Xn - xibarn[[1]]);
    Un = {combineInterps[Un[[1, 1]], T, 1/100],
      combineInterps[Un[[2, 1]], T, 1/100]};
    Print["γ: ", γ];
   ];
   Return[{xibarn, {Xn, Un}}];
  ];

ϵ = 10^-2;
i = 0;
norm_i = 100;

(* Full Algorithm *)
While[
  And[Abs[norm_i] > ϵ, i < 30],
  A = Ta[Asym] /.
    {θ1[t] → xi_i[[1, 1, 1]], θ2[t] → xi_i[[1, 2, 1]], θ1'[t] → xi_i[[1, 3, 1]],
     θ2'[t] → xi_i[[1, 4, 1]], u1[t] → xi_i[[2, 1, 1]], u2[t] → xi_i[[2, 2, 1]]};
```

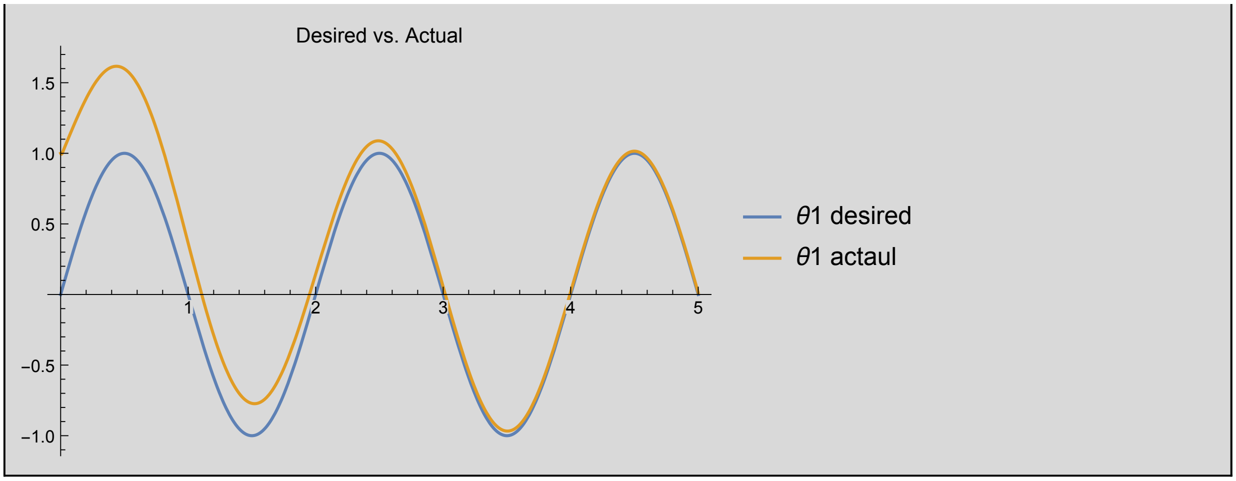
```
A = {Flatten[{combineInterps[A[[1, 1]], T, 1/100],
     combineInterps[A[[1, 2]], T, 1/100], combineInterps[A[[1, 3]], T, 1/100],
     combineInterps[A[[1, 4]], T, 1/100]}],
   Flatten[{combineInterps[A[[2, 1]], T, 1/100],
     combineInterps[A[[2, 2]], T, 1/100], combineInterps[A[[2, 3]], T, 1/100],
     combineInterps[A[[2, 4]], T, 1/100]}],
   Flatten[{combineInterps[A[[3, 1]], T, 1/100],
     combineInterps[A[[3, 2]], T, 1/100], combineInterps[A[[3, 3]], T, 1/100],
     combineInterps[A[[3, 4]], T, 1/100]}],
   Flatten[{combineInterps[A[[4, 1]], T, 1/100],
     combineInterps[A[[4, 2]], T, 1/100], combineInterps[A[[4, 3]], T, 1/100],
     combineInterps[A[[4, 4]], T, 1/100]}]};
B = Ta[Bsym] /. {θ2[t] → xi_i[[1, 2, 1]]};
B = {Flatten[{combineInterps[B[[1, 1]], T, 1/100],
     combineInterps[B[[1, 2]], T, 1/100]}],
   Flatten[{combineInterps[B[[2, 1]], T, 1/100], combineInterps[
      B[[2, 2]], T, 1/100]}], Flatten[{combineInterps[B[[3, 1]], T, 1/100],
     combineInterps[B[[3, 2]], T, 1/100]}], Flatten[{combineInterps[
      B[[4, 1]], T, 1/100], combineInterps[B[[4, 2]], T, 1/100]}]};
a = asym /. {θ1[t] → xi_i[[1, 1, 1]], θ2[t] → xi_i[[1, 2, 1]],
    θ1'[t] → xi_i[[1, 3, 1]], θ2'[t] → xi_i[[1, 4, 1]]};
b = bsym /. {u1[t] → xi_i[[2, 1, 1]], u2[t] → xi_i[[2, 2, 1]]};
Pn = Psol[A, B, Q_n, R_n, P1_n];
rn = rsol[A, B, a, b, Pn, R_n, P1_n, xi_i];
z = zsol[A, B, b, Pn, rn];
v = -Inverse[R_n].(b + B^T.Pn.z + B^T.rn);
v =
 {combineInterps[v[[1, 1]], T, 1/100], combineInterps[v[[2, 1]], T, 1/100]};
zeta = {z, v};
Pr = Psol[A, B, Q_r, R_r, P1_r];
κ = -Inverse[R_r].B^T.Pr;
κ = {Flatten[{combineInterps[κ[[1, 1]], T, 1/100],
     combineInterps[κ[[1, 2]], T, 1/100], combineInterps[
      κ[[1, 3]], T, 1/100], combineInterps[κ[[1, 4]], T, 1/100]}],
   Flatten[{combineInterps[κ[[2, 1]], T, 1/100], combineInterps[
      κ[[2, 2]], T, 1/100], combineInterps[κ[[2, 3]], T, 1/100],
     combineInterps[κ[[2, 4]], T, 1/100]}]};
{xibar_{i+1}, xi_{i+1}} = Armijo[xi_i, xibar_i, zeta, κ];
xibar_{i+1} = {{combineInterps[xibar_{i+1}[[1, 1, 1]], T, 1/100],
    combineInterps[xibar_{i+1}[[1, 2, 1]], T, 1/100],
    combineInterps[xibar_{i+1}[[1, 3, 1]], T, 1/100],
    combineInterps[xibar_{i+1}[[1, 4, 1]], T, 1/100]},
   {combineInterps[xibar_{i+1}[[2, 1, 1]], T, 1/100],
```

```
      combineInterps[xibar_{i+1}[[2, 2, 1]], T, 1/100]}};
   xi_{i+1} = {{combineInterps[xi_{i+1}[[1, 1, 1]], T, 1/100],
      combineInterps[xi_{i+1}[[1, 2, 1]], T, 1/100], combineInterps[
       xi_{i+1}[[1, 3, 1]], T, 1/100], combineInterps[xi_{i+1}[[1, 4, 1]], T, 1/100]},
     {combineInterps[xi_{i+1}[[2, 1, 1]], T, 1/100],
      combineInterps[xi_{i+1}[[2, 2, 1]], T, 1/100]}};
   norm_{i+1} = (DJzeta[xi_i, zeta])[[1, 1]];
   (*Print[xi_{i+1}[[1]]/.t→ T];*)
   Print["norm: ", norm_{i+1}];
   i = i + 1;
   Clear[A, B, a, b, Pn, rn, z, v, zeta, Pr, κ]
  ];

Print["Number of iterations: ", i];
(* Plot the trajectories and control effort *)
Plot[{θ1d[t], xi_i[[1, 1, 1]]}, {t, 0, T}, PlotRange → Full ,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"θ1 desired", "θ1 actaul"}]
Plot[{θ2d[t], xi_i[[1, 2, 1]]}, {t, 0, T}, PlotRange → Full ,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"θ2 desired", "θ2 actaul"}]
Plot[{dθ1d[t], xi_i[[1, 3, 1]]}, {t, 0, T},
 PlotRange → Full , PlotLabel → "Desired vs. Actual",
 PlotLegends → {"θ1' desired", "θ1' actaul"}]
Plot[{dθ2d[t], xi_i[[1, 4, 1]]}, {t, 0, T},
 PlotRange → Full , PlotLabel → "Desired vs. Actual",
 PlotLegends → {"θ2' desired", "θ2' actaul"}]
Plot[{u1d[t], xi_i[[2, 1, 1]]}, {t, 0, T}, PlotRange → Full,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"u1 desired ", "u1 actual"}]
Plot[{u2d[t], xi_i[[2, 2, 1]]}, {t, 0, T}, PlotRange → Full,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"u2 desired ", "u2 actual"}]
ListLinePlot[Table[{h, Abs[norm_h]}, {h, 1, i}], Filling → Axis]

(*ANIMATION*)
(*x and y coordinates for pendulum 1 *)
X1[τ_] := ll1 * Sin[xi_i[[1, 1, 1]]] /. t → τ;
Y1[τ_] := -ll1 * Cos[xi_i[[1, 1, 1]]] /. t → τ;
(*x and y coordinates for pendulum 2*)
X2[τ_] :=
  ll1 * Sin[xi_i[[1, 1, 1]]] + ll2 * Sin[xi_i[[1, 1, 1]] + xi_i[[1, 2, 1]]] /. t → τ;
Y2[τ_] := -ll1 * Cos[xi_i[[1, 1, 1]]] -
    ll2 * Cos[xi_i[[1, 1, 1]] + xi_i[[1, 2, 1]]] /. t → τ;

Animate[Show [Graphics[{Purple, Thick, Line[{{0, 0}, {X1[τ], Y1[τ]}}],
    Purple, Thick, Line[{{X1[τ], Y1[τ]}, {X2[τ], Y2[τ]}}],
```

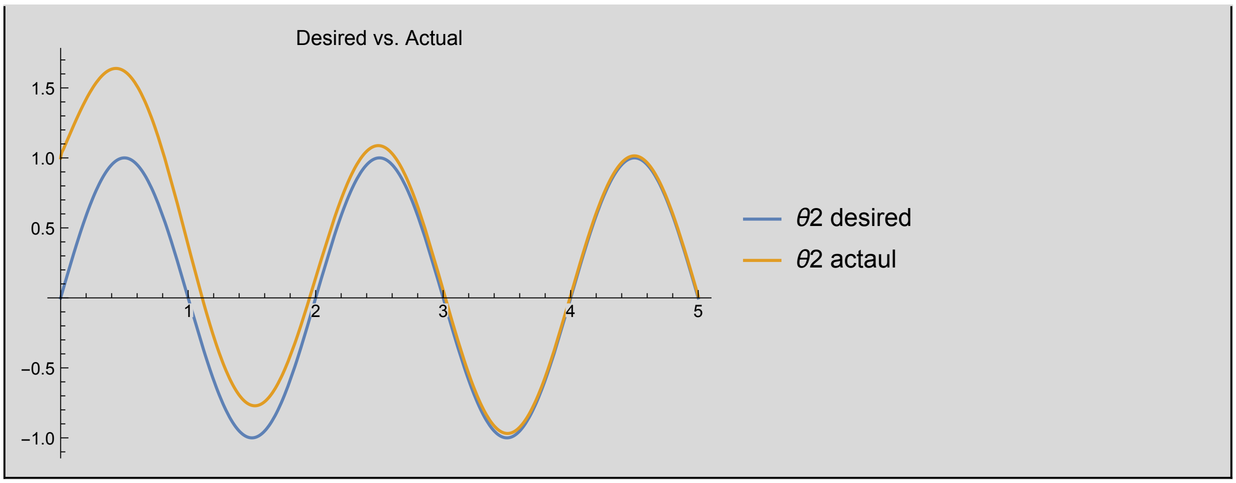
```
    Black, Line[{{-3, 0}, {3, 0}}], Line[{{-3, 0}, {-3, -2}}]}],
   AspectRatio → Automatic, PlotRange → {{-2.1, 2.1}, {-2.1, 2.1}},
   Frame → True], {τ, 0, 2}, AnimationRate → 1]

(*The trajectory plot of the Double Pendulum*)
ParametricPlot[{{X1[τ], Y1[τ]}, {X2[τ], Y2[τ]}},
 {τ, 0, T}, AspectRatio → Automatic, AxesLabel → {x, y},
 PlotLegends → {"Trajectory of pendulum 1", "Trajectory of pendulum 2"}]
```

```
γ: 0.5
norm: -597.39
norm: -148.774
γ: 0.5
γ: 0.25
γ: 0.125
γ: 0.0625
γ: 0.03125
γ: 0.015625
γ: 0.0078125
γ: 0.00390625
γ: 0.00195313
γ: 0.000976563
norm: 0.0471528
γ: 0.5
γ: 0.25
γ: 0.125
γ: 0.0625
γ: 0.03125
γ: 0.015625
γ: 0.0078125
γ: 0.00390625
γ: 0.00195313
γ: 0.000976563
norm: 0.00837522
Number of iterations: 4
```

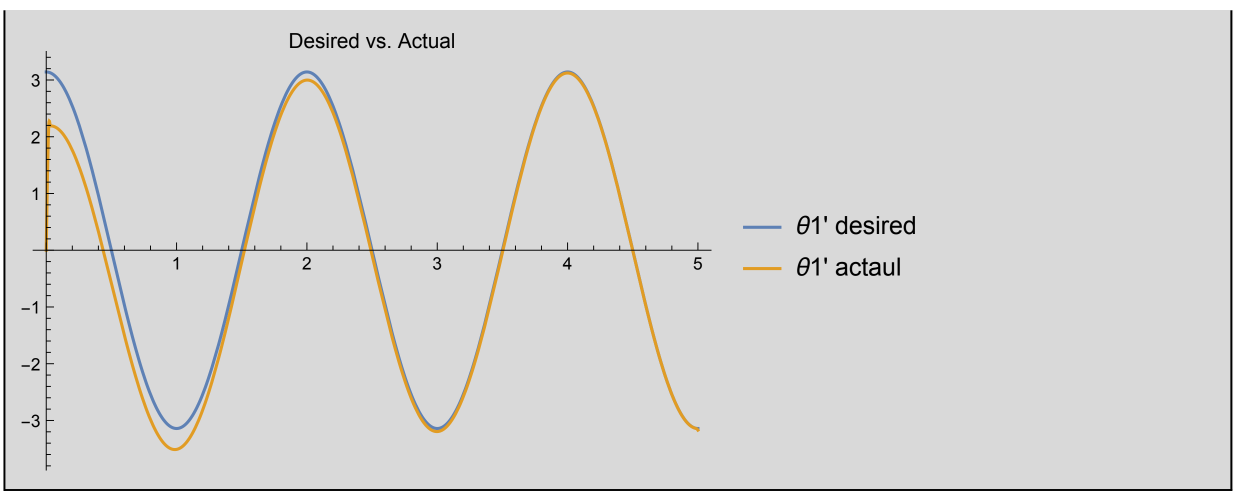
Out[51]=



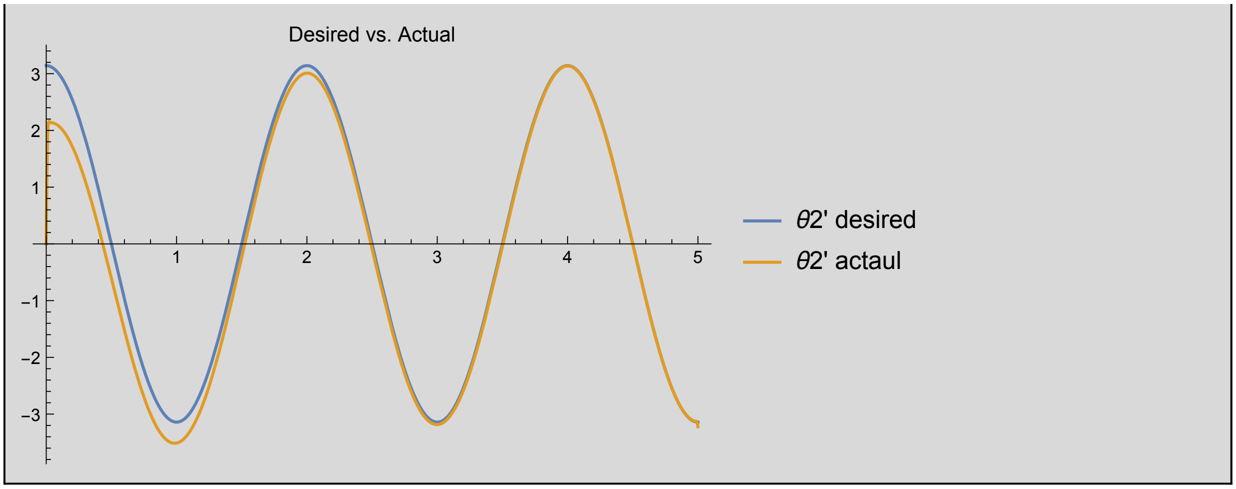
Out[52]=



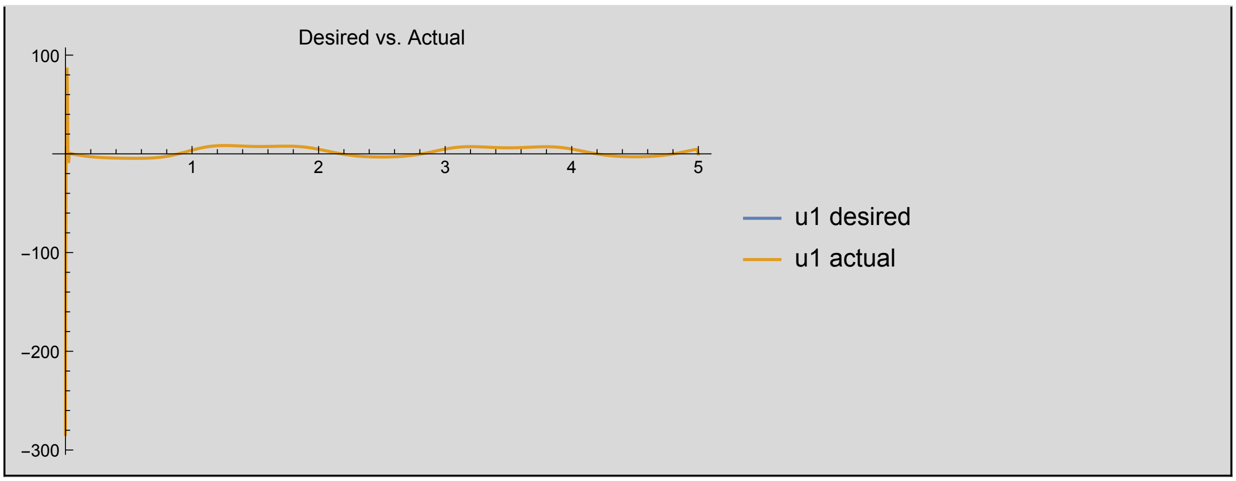
Out[53]=



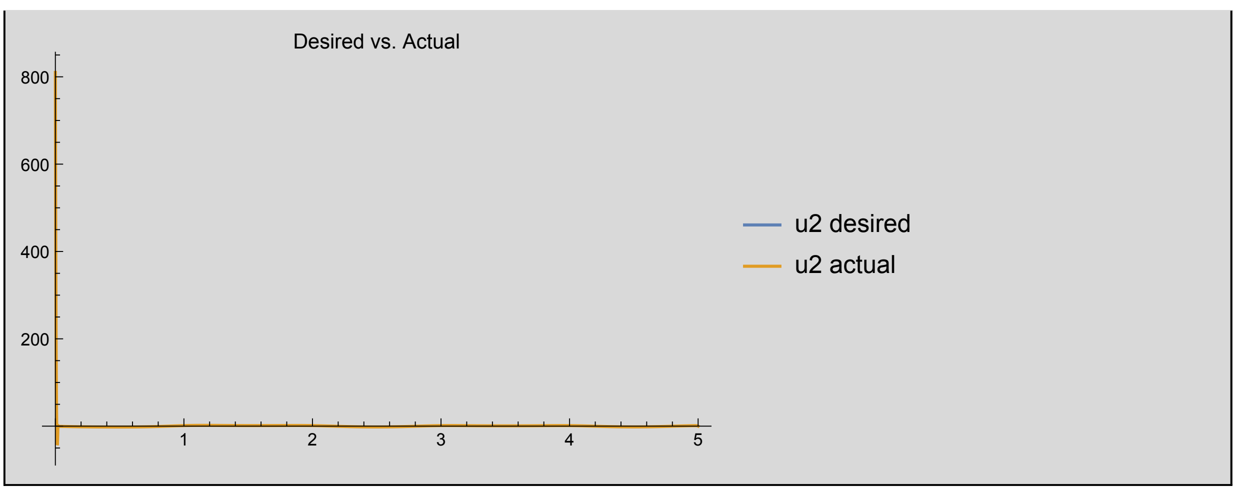
Out[54]=



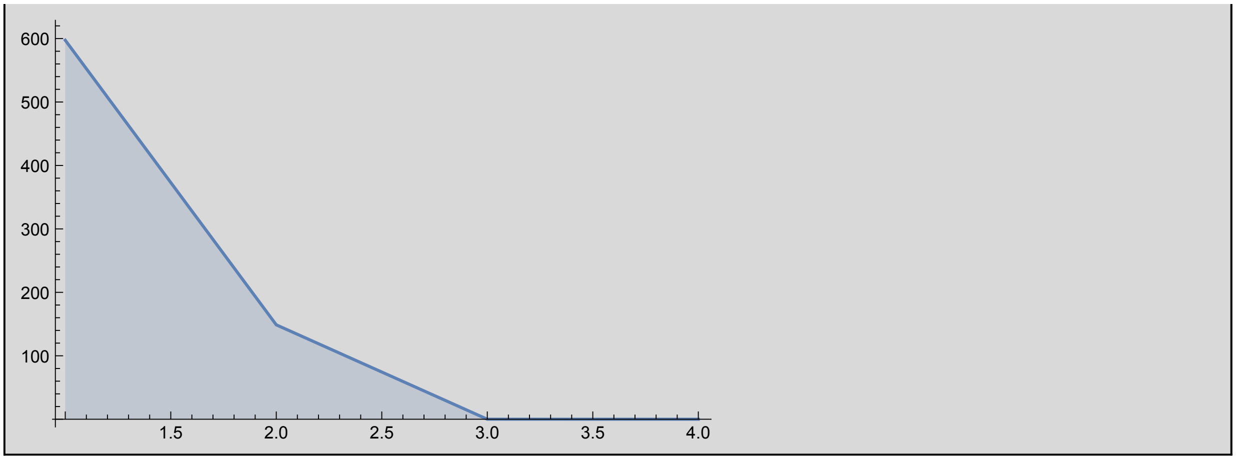
Out[55]=



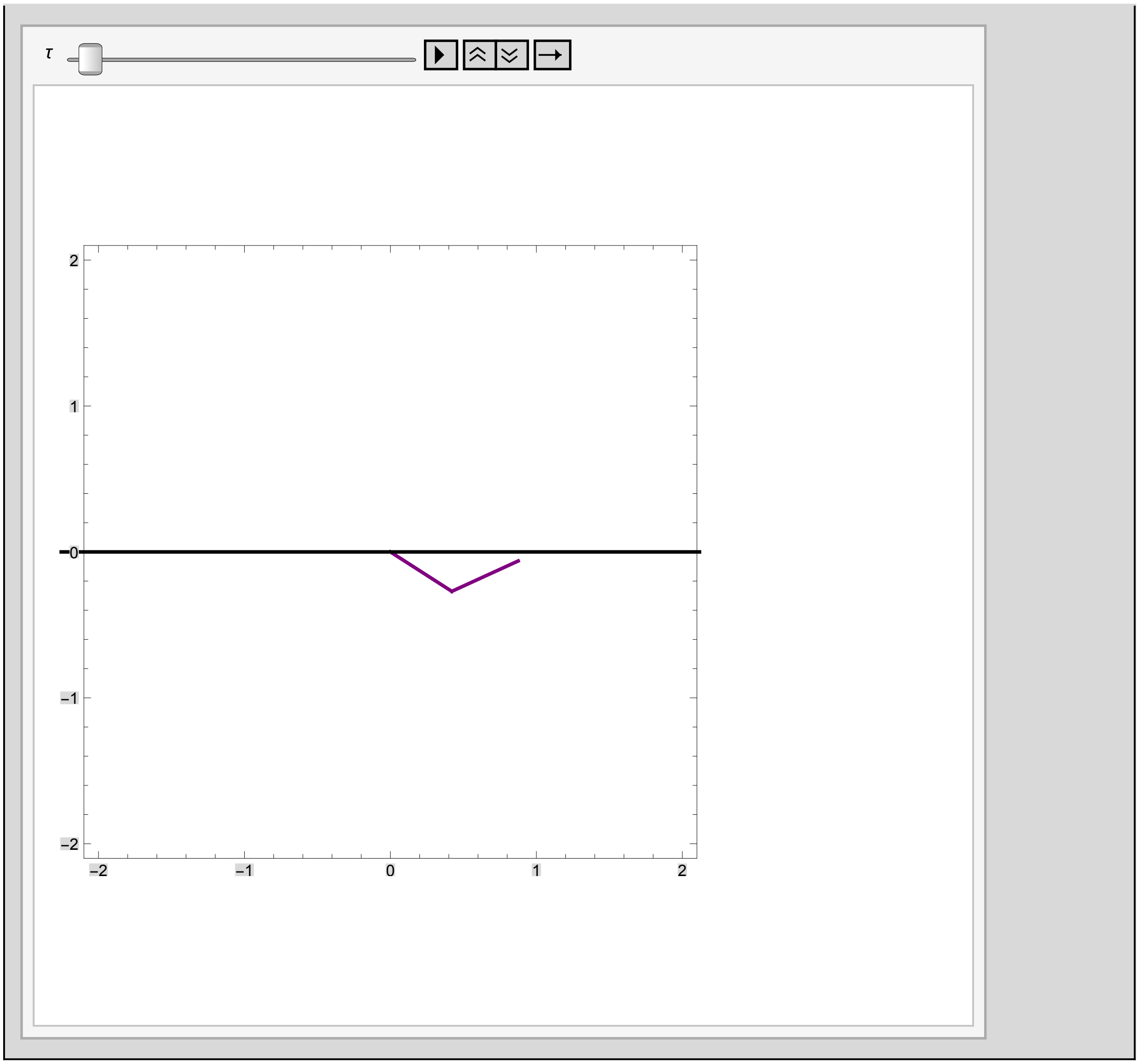
Out[56]=



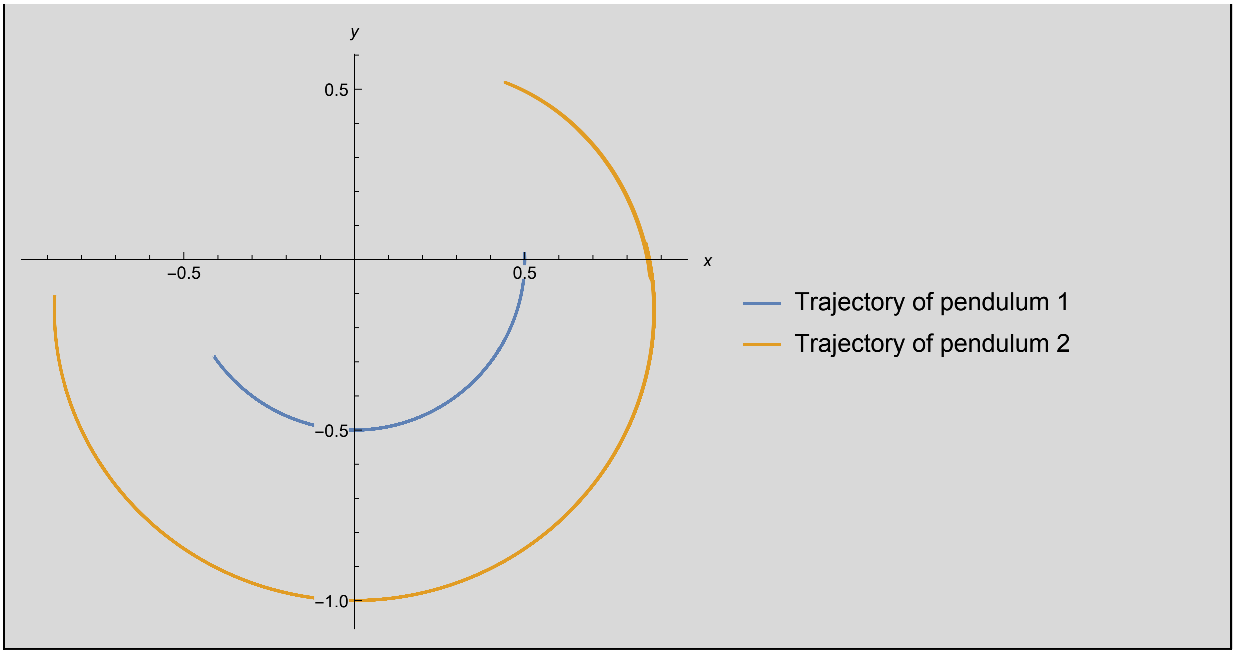
Out[57]=



Out[62]=



Out[63]=



Circular Trajectory

In[102]:=

```
g = 9.81;
T = 20;
l = 1.23;
LL = 1;
m = 1;
h = 0.5;
ρ = 1;

(* State and Control *)
X = {{θ1[t]}, {θ2[t]}, {θ1'[t]}, {θ2'[t]}};
dX = D[X, {t, 1}];
U = {{u1[t]}, {u2[t]}};

(* Desired Trajectory *)
θ1d[t_] := π * t;
θ2d[t_] := -ArcCos[l^2 - LL^2 - LL^2 / (2 * LL * LL)];
dθ1d[t_] := π;
dθ2d[t_] := 0;
(*θ1d[t_]:= Sin[π t];
θ2d[t_]:=Sin[t];
dθ1d[t_]:=π Cos[π t];
dθ2d[t_]:=π Cos[π t];*)
```

```
Xd[t_] := {{θ1d[t]}, {θ2d[t]}, {dθ1d[t]}, {dθ2d[t]}};

(* Adding Obstacel *)
Obs = {{1.25}, {1.25}};

Q = 10 * IdentityMatrix[4];
Qn = Q;
Qr = Q;

R = 0.001 * IdentityMatrix[2];
Rn = R;
Rr = R;

P1 = P1 = 0.1 * IdentityMatrix[4];
P1n = P1;
P1r = P1;

L[X_, U_] := 1/2 ((X - Xd[t])ᵀ.Q.(X - Xd[t])) + 1/2 Uᵀ.R.U ;
J[X_, U_] := Quiet[NIntegrate[L[X, U], {t, 0, T},
      Method → {Automatic, "SymbolicProcessing" → False}]] +
   1/2 ((X - Xd[t]) /. t → T)ᵀ.P1.((X - Xd[t]) /. t → T);

(*Use double pendulum anastasia hw3*)
f[x_, u_] := {{x[[3, 1]]}, {x[[4, 1]]},
   {(3 * (u[[1, 1]] - 2 (-g * LL * m * (-2 * Sin[x[[1, 1]]] + Sin[x[[1, 1]] + x[[2, 1]]]) +
              u[[2, 1]]))) / (4 * LL * (3 * LL * m + 2 * h^4 * ρ + 8 * h^2 * LL^2 * ρ))},
    {(3 * (2 * g * m * (8 * h^2 * LL * (h^2 + 4 * LL^2) * ρ * Sin[x[[1, 1]]] -
              3 * LL * (2 * h^4 * ρ + LL * (m + 8 * h^2 * LL * ρ)) Sin[x[[1, 1]] + x[[2, 1]]]) -
           4 * h^2 * (h^2 + 4 * LL^2) * ρ * u[[1, 1]] +
           (3 * LL * m + 10 * h^2 (h^2 + 4 * LL^2) * ρ) * u[[2, 1]])) /
      (8 * h^2 * LL * (h^2 + 4 * LL^2) * ρ * (3 * LL * m + 2 * h^4 * ρ + 8 * h^2 * LL^2 * ρ))
   }};


DJzeta[xi_, zeta_] :=
  Module[{X = xi[[1]], U = xi[[2]], z = zeta[[1]], v = zeta[[2]]},
   Return[Quiet[NIntegrate[(Q.(X - Xd[t]))ᵀ.z + (R.U)ᵀ.v, {t, 0, T},
         Method → {Automatic, "SymbolicProcessing" → False}]] +
       ((P1.(X - Xd[t]))ᵀ.z) /. t → T];
  ];

xibar0 = {Xd[t], {{0}, {0}}};
```

```
xi0 = {{{1}, {1}, {0}, {0}}, {{0}, {0}}};

(* Symbolic forms of A, B, a, b. Ta transforms A and B to the proper size *)
Asym = D[(f[X, U]), X^T];
(*Print["A: ", Asym]*)
Bsym = D[f[X, U], U^T];
(*Print["B: ", Bsym]*)
asym = D[L[X, U], {X, 1}][[1, 1]];
bsym = D[L[X, U], {U, 1}][[1, 1]];
Ta[R_] := Table[R[[i, 1]], {i, 1, 4}];



(* Riccati solution for P *)
Psol[A_, B_, Q_ , R_, P1_] := Module[{PEQ1, PEQ2, Ps, i, j},
   Ri[t_] := Table[P_{i,j}[t], {i, 1, 4}, {j, 1, 4}];
   PEQ1 = (Ri'[t] + A^T.Ri[t] + Ri[t].A - Ri[t].B.Inverse[R].B^T.Ri[t] + Q) ==
     {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}};
   PEQ2 = Ri[T] == P1;
   Ps = (NDSolve[{PEQ1, PEQ2}, Flatten[Ri[t]], {t, 0, T}])[[1]];
   Return[Ri[t] /. Ps];
  ];

(* Riccati solution for r *)
rsol[A_, B_, a_, b_, P_, R_, P1_, xi_] := Module[{rEQ1, rEQ2, rs},
   Rir[t_] := {{r1[t]}, {r2[t]}, {r3[t]}, {r4[t]}};
   rEQ1 =
    Rir'[t] + (A - B.Inverse[R].B^T.P)^T.Rir[t] + a - P.B.Inverse[R].b ==
     {{0}, {0}, {0}, {0}};
   rEQ2 = Rir[T] == (P1.(xi[[1]] - Xd[t])) /. t → T;
   rs = (NDSolve[{rEQ1, rEQ2}, Flatten[Rir[t]], {t, 0, T}])[[1]];
   Return[Rir[t] /. rs];
  ];

(* Descent Direction solution *)
zsol[A_, B_, b_, P_, r_] := Module[{v, zEQ1, zEQ2, zs},
   v = -Inverse[Rn].(b + B^T.P.{{z1[t]}, {z2[t]}, {z3[t]}, {z4[t]}} + B^T.r);
   zEQ1 = {{z1'[t]}, {z2'[t]}, {z3'[t]}, {z4'[t]}} ==
     A.{{z1[t]}, {z2[t]}, {z3[t]}, {z4[t]}} + B.v;
   zEQ2 = {{z1[0]}, {z2[0]}, {z3[0]}, {z4[0]}} == {{0}, {0}, {0}, {0}};
   (* zEQ2 = {{z1[0]},{z2[0]}}== -Inverse[P... *)
   zs = (NDSolve[{zEQ1, zEQ2}, {z1[t], z2[t], {z3[t]}, {z4[t]}}, {t, 0, T}])[[
     1]];
   Return[({{z1[t]}, {z2[t]}, {z3[t]}, {z4[t]}} /. zs)];
```

```
];

(* Projection of xibar onto feasible space *)
Proj[xibar_, K_] :=
  Module[{xbar = xibar[[1]], ubar = xibar[[2]], xEQ1, xEQ2, xs},
   xEQ1 = {dX[[3]], dX[[4]]} == {(f[X, U] /. {u1[t] → (ubar + K.(X - xbar))[[1, 1]],
          u2[t] → (ubar + K.(X - xbar))[[2, 1]]})[[3]],
      (f[X, U] /. {u1[t] → (ubar + K.(X - xbar))[[1, 1]],
          u2[t] → (ubar + K.(X - xbar))[[2, 1]]})[[4]]};
   xEQ2 = {{θ1[0]}, {θ2[0]}, {θ1'[0]}, {θ2'[0]}} == {{1}, {1}, {0}, {0}};
   xs =
    (NDSolve[{xEQ1, xEQ2}, {θ1[t], θ2[t], θ1'[t], θ2'[t]}, {t, 0, T}])[[1]];
   Return[(X /. xs)];
  ];

combineInterps[interp_, maxIndex_, stepSize_] :=
  Module[{samples, index, val, retInterp}, samples = {};
   index = 0;
   While[index ≤ maxIndex, AppendTo[samples, {index, interp /. t → index}];
    index += stepSize;];
   Return[{Interpolation[samples, Method → "Hermite"][t]}];];

(* Armijo Line Search *)
Armijo[xi_, xibar_, zeta_, K_, maxIters_: 10] :=
  Module[{α = .0001, β = .5, n = 0, xibarn, γ, X, U, Xn, Un, Jtemp, DJtemp},
   X = xi[[1]];
   U = xi[[2]];
   γ = β^n;
   xibarn = xibar + γ zeta;
   Xn  = Proj[xibarn, K];
   Un  = xibarn[[2]] + K.(Xn - xibarn[[1]]);
   Un = {combineInterps[Un[[1, 1]], T, 1/100],
     combineInterps[Un[[2, 1]], T, 1/100]};
   Jtemp = J[X, U]; (* Only changes in main loop *)
   DJtemp = DJzeta[xi, zeta]; (* Only changes in main loop *)
   While[
    And[(J[Xn, Un])[[1, 1]] > (Jtemp + α γ DJtemp )[[1, 1]], n < maxIters],
    n = n + 1;
    γ = β^n;
    xibarn = xibar + γ zeta;
    Xn = Proj[xibarn, K];
    Un = xibarn[[2]] + K.(Xn - xibarn[[1]]);
    Un = {combineInterps[Un[[1, 1]], T, 1/100],
```

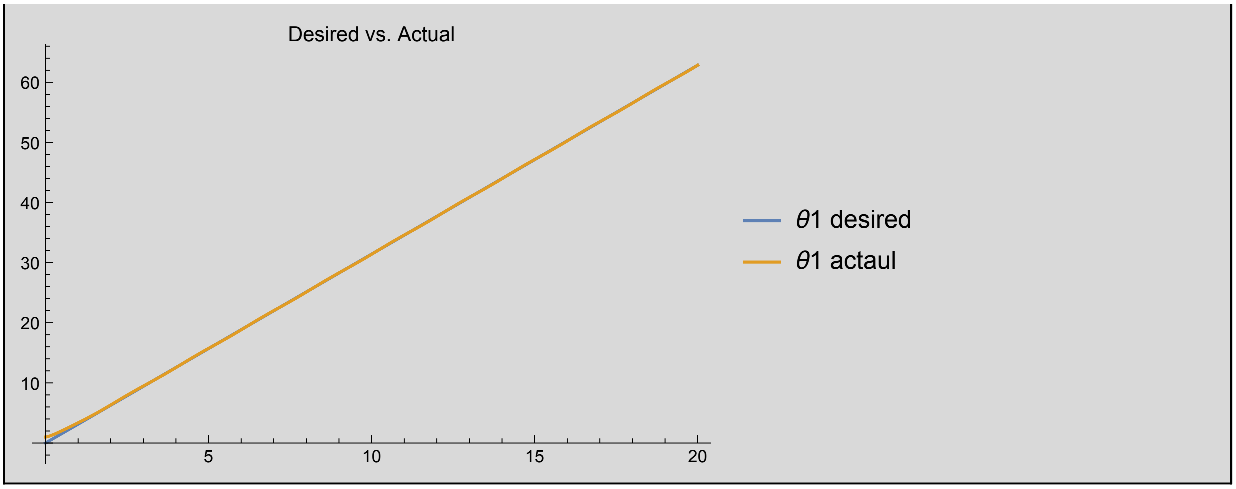
```
       combineInterps[Un[[2, 1]], T, 1/100]};
     Print["n: ", n];
    ];
   Return[{xibarn, {Xn, Un}}];
  ];

ϵ = 10^-2;
i = 0;
normi = 100;

(* Full Algorithm *)
While[
  And[Abs[normi] > ϵ, i < 30],
  A = Ta[Asym] /.
    {θ1[t] → xii[[1, 1, 1]], θ2[t] → xii[[1, 2, 1]], θ1'[t] → xii[[1, 3, 1]],
     θ2'[t] → xii[[1, 4, 1]], u1[t] → xii[[2, 1, 1]], u2[t] → xii[[2, 2, 1]]};
  B = Ta[Bsym] /. {θ2[t] → xii[[1, 2, 1]]};
  a = asym /. {θ1[t] → xii[[1, 1, 1]], θ2[t] → xii[[1, 2, 1]],
     θ1'[t] → xii[[1, 3, 1]], θ2'[t] → xii[[1, 4, 1]]};
  b = bsym /. {u1[t] → xii[[2, 1, 1]], u2[t] → xii[[2, 2, 1]]};
  Pn = Psol[A, B, Qn, Rn, P1n];
  rn = rsol[A, B, a, b, Pn, Rn, P1n, xii];
  z = zsol[A, B, b, Pn, rn];
  v = -Inverse[Rn].(b + B^T.Pn.z + B^T.rn);
  v =
   {combineInterps[v[[1, 1]], T, 1/100], combineInterps[v[[2, 1]], T, 1/100]};
  zeta = {z, v};
  Pr = Psol[A, B, Qr, Rr, P1r];
  κ = -Inverse[Rr].B^T.Pr;
  κ = {Flatten[{combineInterps[κ[[1, 1]], T, 1/100],
      combineInterps[κ[[1, 2]], T, 1/100], combineInterps[
       κ[[1, 3]], T, 1/100], combineInterps[κ[[1, 4]], T, 1/100]}],
    Flatten[{combineInterps[κ[[2, 1]], T, 1/100], combineInterps[
       κ[[2, 2]], T, 1/100], combineInterps[κ[[2, 3]], T, 1/100],
      combineInterps[κ[[2, 4]], T, 1/100]}]};
  {xibari+1, xii+1} = Armijo[xii, xibari, zeta, κ];
  xibari+1 = {{combineInterps[xibari+1[[1, 1, 1]], T, 1/100],
     combineInterps[xibari+1[[1, 2, 1]], T, 1/100],
     combineInterps[xibari+1[[1, 3, 1]], T, 1/100],
     combineInterps[xibari+1[[1, 4, 1]], T, 1/100]},
    {combineInterps[xibari+1[[2, 1, 1]], T, 1/100],
     combineInterps[xibari+1[[2, 2, 1]], T, 1/100]}};
  xii+1 = {{combineInterps[xii+1[[1, 1, 1]], T, 1/100],
```

```
      combineInterps[xi_{i+1}[[1, 2, 1]], T, 1/100], combineInterps[
       xi_{i+1}[[1, 3, 1]], T, 1/100], combineInterps[xi_{i+1}[[1, 4, 1]], T, 1/100]},
     {combineInterps[xi_{i+1}[[2, 1, 1]], T, 1/100],
      combineInterps[xi_{i+1}[[2, 2, 1]], T, 1/100]}};
   norm_{i+1} = (DJzeta[xi_i, zeta])[[1, 1]];
   Print[xi_{i+1}[[1]] /. t → T];
   Print[norm_{i+1}];
   i = i + 1;
   Clear[A, B, a, b, Pn, rn, z, v, zeta, Pr, κ]
  ];


Print["Number of iterations: ", i];
(* Plot the trajectories and control effort *)
Plot[{θ1d[t], xi_i[[1, 1, 1]]}, {t, 0, T}, PlotRange → Full ,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"θ1 desired", "θ1 actaul"}]
Plot[{θ2d[t], xi_i[[1, 2, 1]]}, {t, 0, T}, PlotRange → Full ,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"θ2 desired", "θ2 actaul"}]
Plot[{dθ1d[t], xi_i[[1, 3, 1]]}, {t, 0, T},
 PlotRange → Full , PlotLabel → "Desired vs. Actual",
 PlotLegends → {"θ1' desired", "θ1' actaul"}]
Plot[{dθ2d[t], xi_i[[1, 4, 1]]}, {t, 0, T},
 PlotRange → Full , PlotLabel → "Desired vs. Actual",
 PlotLegends → {"θ2' desired", "θ2' actaul"}]
Plot[{u1d[t], xi_i[[2, 1, 1]]}, {t, 0, T}, PlotRange → Full,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"u1 desired ", "u1 actual"}]
Plot[{u2d[t], xi_i[[2, 2, 1]]}, {t, 0, T}, PlotRange → Full,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"u2 desired ", "u2 actual"}]
ListLinePlot[Table[{h, Abs[norm_h]}, {h, 1, i}], Filling → Axis]

(*ANIMATION*)
(*x and y coordinates for pendulum 1 *)
X1[τ_] := ll1 * Sin[xi_i[[1, 1, 1]]] /. t → τ;
Y1[τ_] := -ll1 * Cos[xi_i[[1, 1, 1]]] /. t → τ;
(*x and y coordinates for pendulum 2*)
X2[τ_] :=
  ll1 * Sin[xi_i[[1, 1, 1]]] + ll2 * Sin[xi_i[[1, 1, 1]] + xi_i[[1, 2, 1]]] /. t → τ;
Y2[τ_] := -ll1 * Cos[xi_i[[1, 1, 1]]] -
    ll2 * Cos[xi_i[[1, 1, 1]] + xi_i[[1, 2, 1]]] /. t → τ;

Animate[Show [Graphics[{Purple, Thick, Line[{{0, 0}, {X1[τ], Y1[τ]}}],
    Purple, Thick, Line[{{X1[τ], Y1[τ]}, {X2[τ], Y2[τ]}}],
    Black, Line[{{-3, 0}, {3, 0}}], Line[{{-3, 0}, {-3, -2}}]}],
```

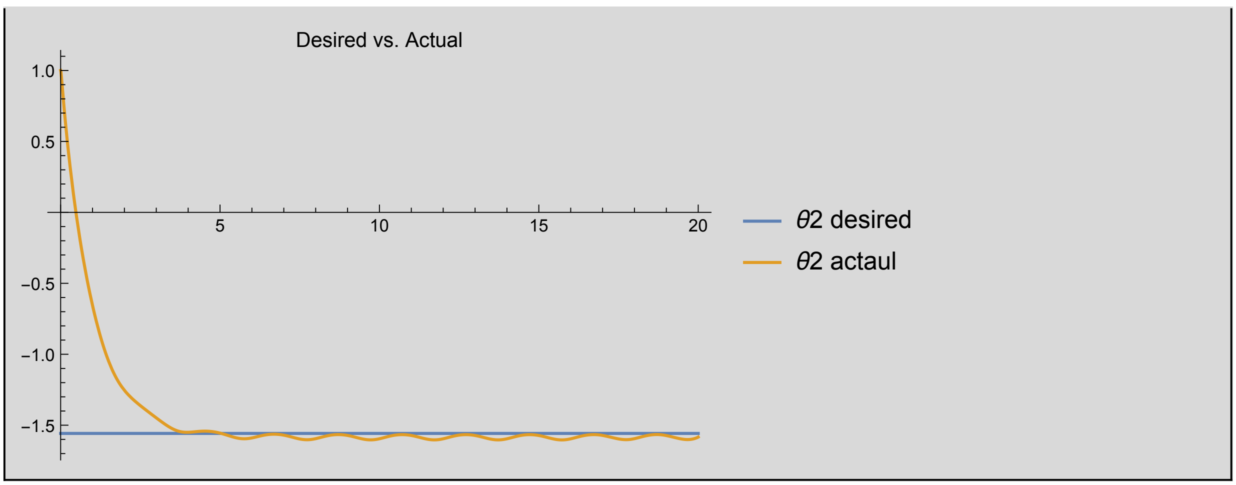
```
  AspectRatio → Automatic, PlotRange → {{-2.1, 2.1}, {-2.1, 2.1}},
  Frame → True], {τ, 0, 2}, AnimationRate → 1]

(*The trajectory plot of the Double Pendulum*)
ParametricPlot[{{X1[τ], Y1[τ]}, {X2[τ], Y2[τ]}},
 {τ, 0, T}, AspectRatio → Automatic, AxesLabel → {x, y},
 PlotLegends → {"Trajectory of pendulum 1", "Trajectory of pendulum 2"}]
```

```
Number of iterations: 22
```

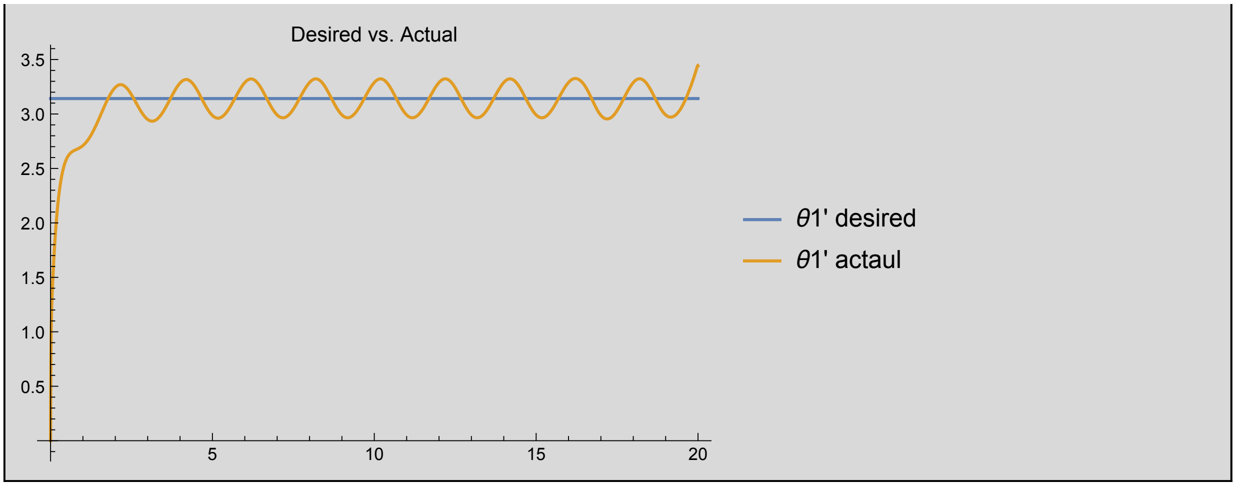
Out[149]=



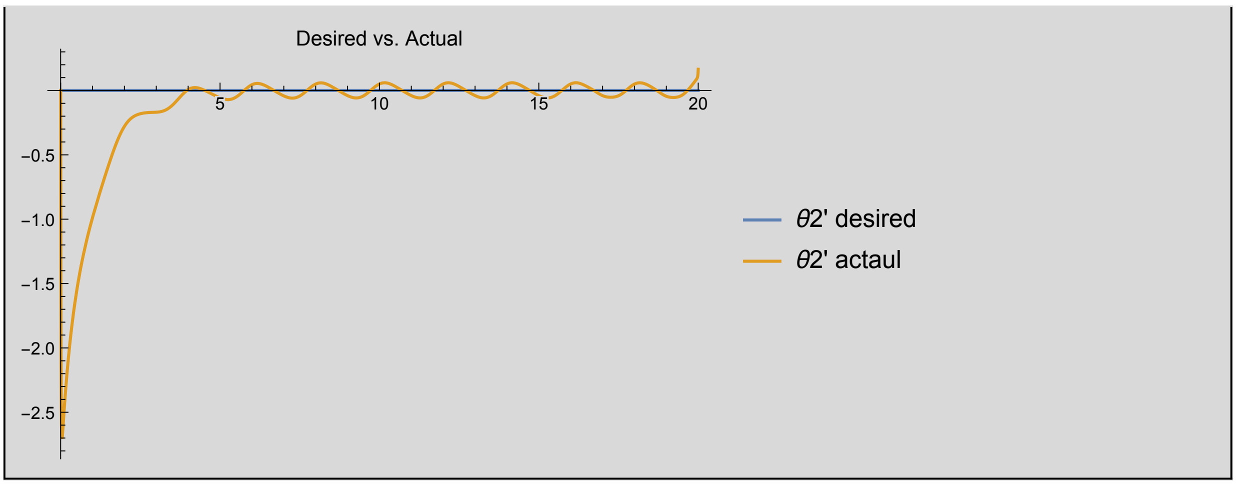
Out[150]=



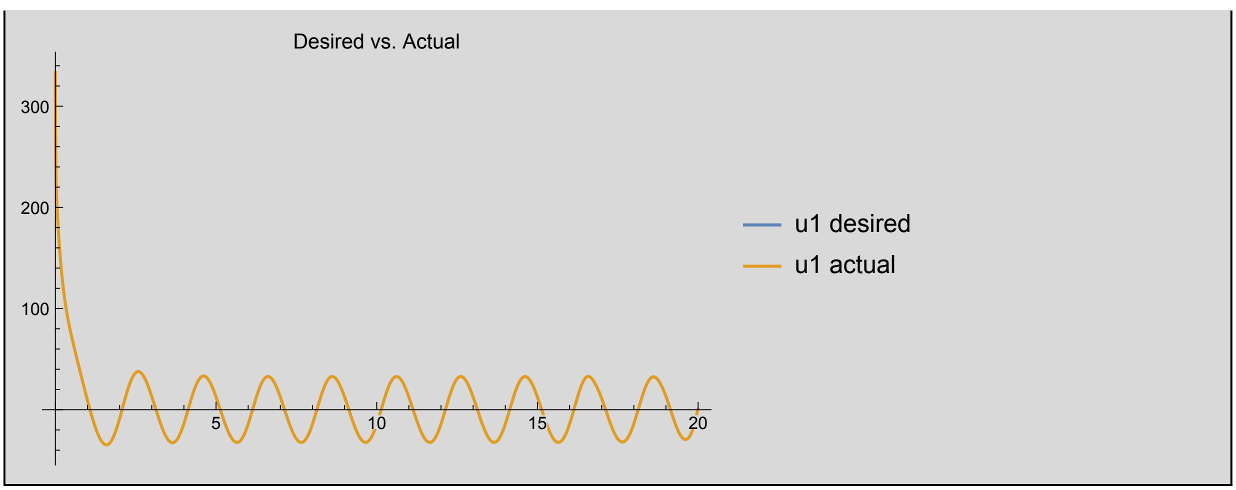
Out[151]=



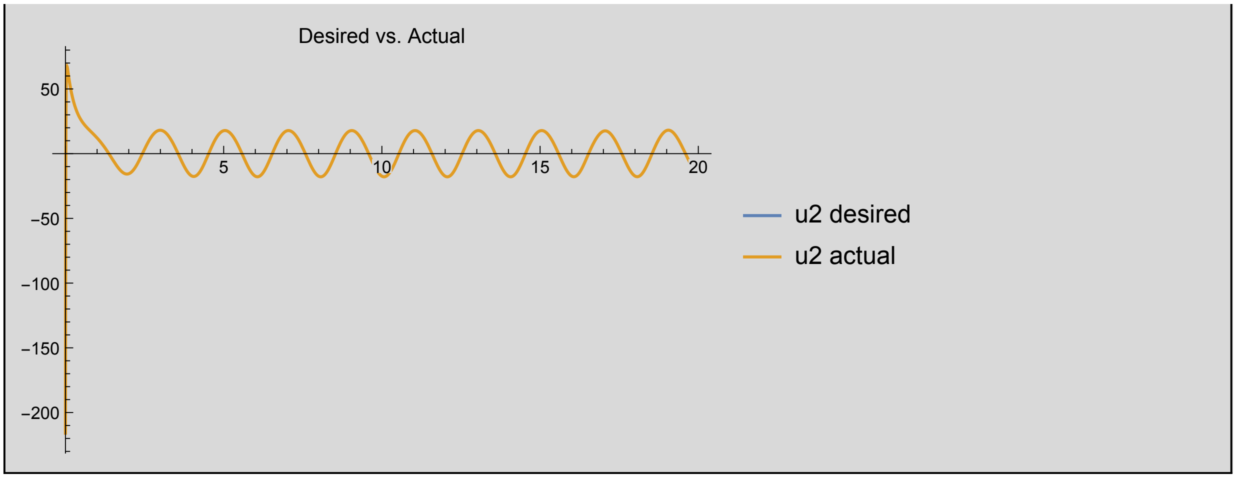
Out[152]=



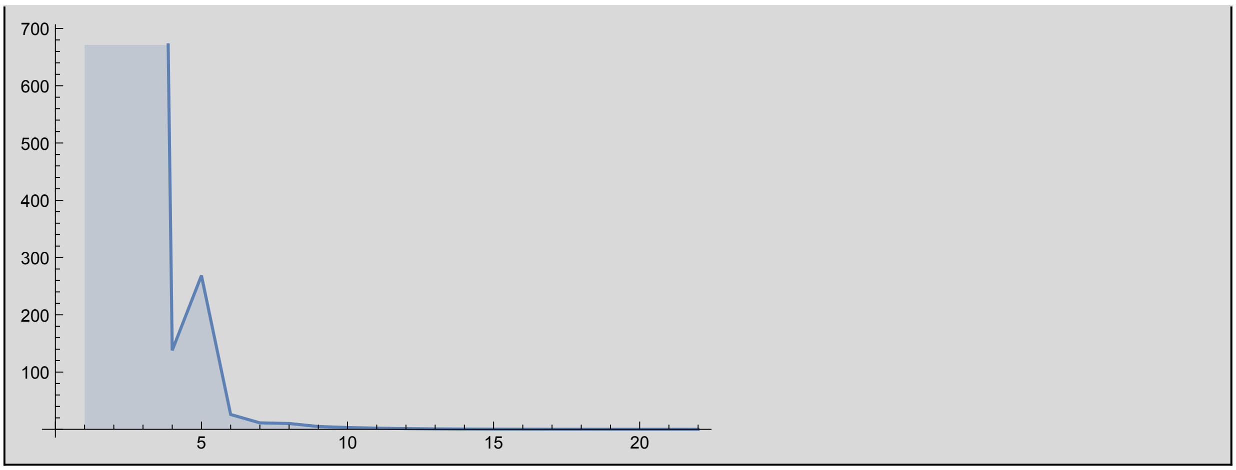
Out[153]=



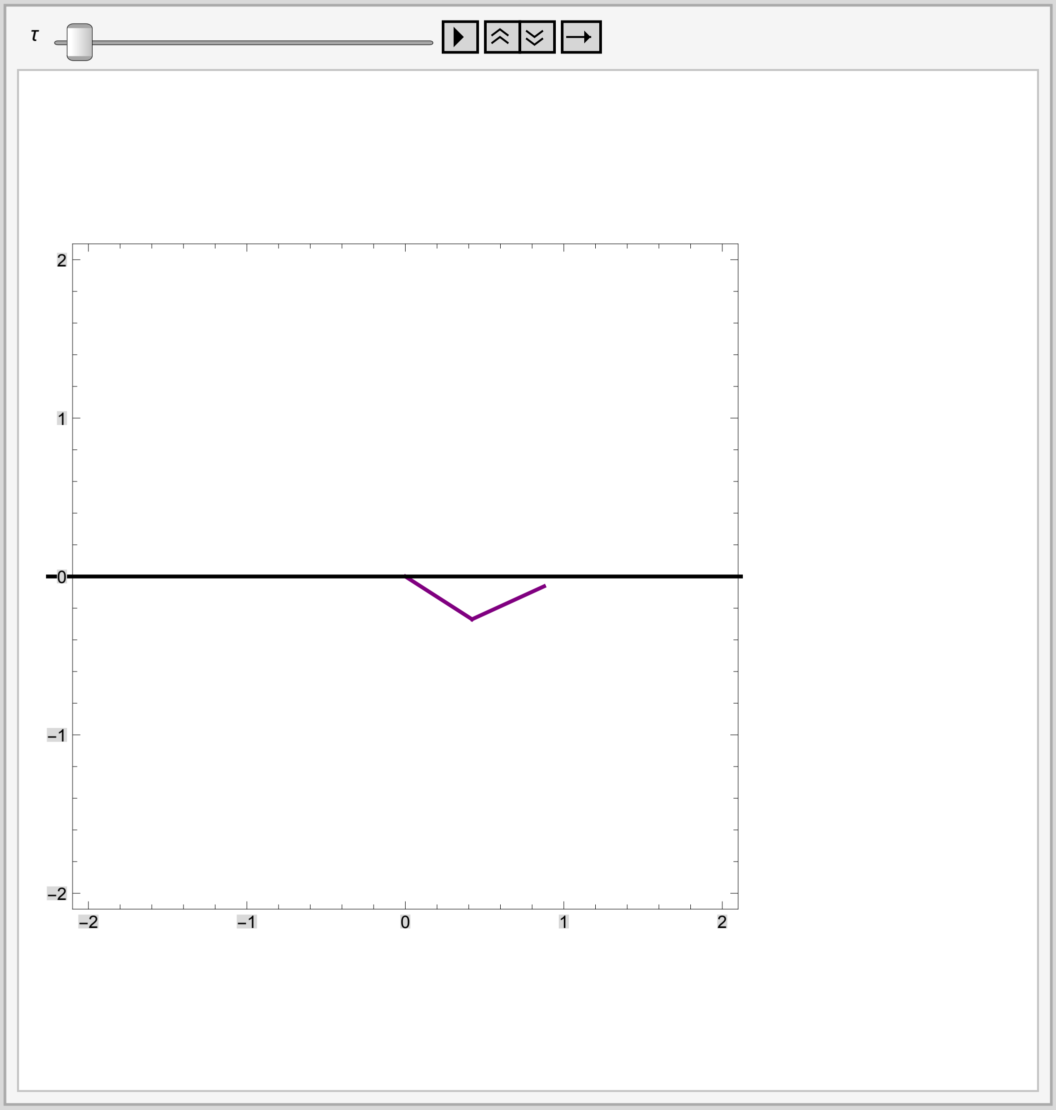
Out[154]=



Out[155]=



Out[160]=



Piecewise Trajectory

In[162]:=

```
g = 9.81;
T = 5;

(* link lenghts *)
ll1 = 0.5;
ll2 = 0.5;
(* mass of each link *)
mm1 = 0.5;
mm2 = 0.5;
(* Distance to center of mass for each link *)
```

```
rr1 = 0.25;
rr2 = 0.25;
(* Damping coefficient of joints *)
bb1 = 0 ;
bb2 = 0;

Iz1 = 0.01;
Iz2 = 0.01;
αα = Iz1 + Iz2 + mm1 rr1^2 + mm2 (ll1^2 + ll2^2);
ββ = mm2 ll2 rr2;
δδ = Iz2 + mm2 rr2^2;

(* State and Control *)
X = {{θ1[t]}, {θ2[t]}, {θ1'[t]}, {θ2'[t]}};
dX = D[X, {t, 1}];
U = {{u1[t]}, {u2[t]}};

(* Desired Trajectory *)
αd = 45 * Pi / 180;
θ1d[t_] := Piecewise[{{1, 0 < t < 1},
     {1 + (αd / 2) * (1 - Cos[(2 Pi / 4) * (t - 3)]), 1 ≤ t < 3}, {1, 3 ≤ t < 5}}];
θ2d[t_] := Piecewise[{{1, 0 < t < 1},
     {1 + (αd / 2) * (1 - Cos[(2 Pi / 4) * (t - 3)]), 1 ≤ t < 3}, {1, 3 ≤ t < 5}}];
dθ1d[t_] := Piecewise[{{0, 0 < t < 1}, {π * (αd / 4) * Sin[π * (t - 3)], 1 ≤ t < 3},
     {0, 3 ≤ t < 5}}];
dθ2d[t_] := Piecewise[{{0, 0 < t < 1}, {π * (αd / 4) * Sin[π * (t - 3)], 1 ≤ t < 3},
     {0, 3 ≤ t < 5}}];
Xd[t_] := {{θ1d[t]}, {θ2d[t]}, {dθ1d[t]}, {dθ2d[t]}};
u1d[t_] := 0;
u2d[t_] := 0;
Ud[t_] := {{u1d[t]}, {u2d[t]}};

Q = 10 * IdentityMatrix[4];
Qn = Q;
Qr = Q;

R = 0.001 * IdentityMatrix[2];
Rn = R;
Rr = R;

P1 = 1 * IdentityMatrix[4];
P1n = 0 * IdentityMatrix[4];
P1r = 0 * IdentityMatrix[4];
```

```
L[X_, U_] := 1/2 ((X - Xd[t])ᵀ.Q.(X - Xd[t])) + 1/2 Uᵀ.R.U;
J[X_, U_] := Quiet[NIntegrate[L[X, U], {t, 0, T},
      Method → {Automatic, "SymbolicProcessing" → False}]] +
   1/2 ((X - Xd[t]) /. t → T)ᵀ.P1.((X - Xd[t]) /. t → T);

(*Use double pendulum anastasia hw3*)
f[x_, u_] := {
   {x[[3, 1]]},
    {x[[4, 1]]}, {1/((αα - δδ) δδ - ββ^2 Cos[x[[2, 1]]]^2)
      (δδ (u[[1, 1]] - u[[2, 1]]) - g (ll1 mm2 + mm1 rr1) δδ Cos[x[[1, 1]]] +
        ββ Cos[x[[2, 1]]] (-u[[2, 1]] + g mm2 rr2 Cos[x[[1, 1]] + x[[2, 1]]]) +
        ββ^2 Cos[x[[2, 1]]] Sin[x[[2, 1]]] x[[3, 1]]^2 + bb2 ββ Cos[x[[2, 1]]]
         x[[4, 1]] + ββ δδ Sin[x[[2, 1]]] (x[[3, 1]] + x[[4, 1]])^2 +
        δδ (-bb1 x[[3, 1]] + bb2 x[[4, 1]]))}, {-1/((αα - δδ) δδ - ββ^2 Cos[x[[2, 1]]]^2)
      (δδ u[[1, 1]] - αα u[[2, 1]] + ββ (u[[1, 1]] - 2 u[[2, 1]]) Cos[x[[2, 1]]] -
        g (ll1 mm2 + mm1 rr1) Cos[x[[1, 1]]] (δδ + ββ Cos[x[[2, 1]]]) +
        g mm2 rr2 (αα - δδ + ββ Cos[x[[2, 1]]]) Cos[x[[1, 1]] + x[[2, 1]]] +
        ββ (αα + 2 ββ Cos[x[[2, 1]]]) Sin[x[[2, 1]]] x[[3, 1]]^2 +
        bb2 (αα + 2 ββ Cos[x[[2, 1]]]) x[[4, 1]] +
        ββ (δδ + ββ Cos[x[[2, 1]]]) Sin[x[[2, 1]]] x[[4, 1]]^2 +
        (δδ + ββ Cos[x[[2, 1]]]) x[[3, 1]] (-bb1 + 2 ββ Sin[x[[2, 1]]] x[[4, 1]]))}
  };


DJzeta[xi_, zeta_] :=
  Module[{X = xi[[1]], U = xi[[2]], z = zeta[[1]], v = zeta[[2]]},
   Return[Quiet[NIntegrate[(Q.(X - Xd[t]))ᵀ.z + (R.U)ᵀ.v, {t, 0, T},
         Method → {Automatic, "SymbolicProcessing" → False}]] +
       ((P1.(X - Xd[t]))ᵀ.z) /. t → T];
  ];

xibar0 = {Xd[t], Ud[t]};
xi0 = {{{1}, {1}, {0}, {0}}, {{0}, {0}}};

(* Symbolic forms of A, B, a, b. Ta transforms A and B to the proper size *)
Asym = D[(f[X, U]), Xᵀ];
(*Print["A: ", Asym]*)
Bsym = D[f[X, U], Uᵀ];
(*Print["B: ", Bsym]*)
asym = D[L[X, U], {X, 1}][[1, 1]];
```

```
bsym = D[L[X, U], {U, 1}][[1, 1]];
Ta[R_] := Table[R[[i, 1]], {i, 1, 4}];



(* Riccati solution for P *)
Psol[A_, B_, Q_ , R_, P1_] := Module[{PEQ1, PEQ2, Ps, i, j},
   Ri[t_] := Table[Pi,j[t], {i, 1, 4}, {j, 1, 4}];
   PEQ1 = (Ri'[t] + AT.Ri[t] + Ri[t].A - Ri[t].B.Inverse[R].BT.Ri[t] + Q) ==
     {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}};
   PEQ2 = Ri[T] == P1;
   Ps = (NDSolve[{PEQ1, PEQ2}, Flatten[Ri[t]], {t, 0, T}])[[1]];
   Return[Ri[t] /. Ps];
  ];

(* Riccati solution for r *)
rsol[A_, B_, a_, b_, P_, R_, P1_, xi_] := Module[{rEQ1, rEQ2, rs},
   Rir[t_] := {{r1[t]}, {r2[t]}, {r3[t]}, {r4[t]}};
   rEQ1 =
    Rir'[t] + (A - B.Inverse[R].BT.P)T.Rir[t] + a - P.B.Inverse[R].b ==
     {{0}, {0}, {0}, {0}};
   rEQ2 = Rir[T] == (P1.(xi[[1]] - Xd[t])) /. t → T;
   rs = (NDSolve[{rEQ1, rEQ2}, Flatten[Rir[t]], {t, 0, T}])[[1]];
   Return[Rir[t] /. rs];
  ];

(* Descent Direction solution *)
zsol[A_, B_, b_, P_, r_] := Module[{v, zEQ1, zEQ2, zs},
   v = -Inverse[Rn].(b + BT.P.{{z1[t]}, {z2[t]}, {z3[t]}, {z4[t]}} + BT.r);
   zEQ1 = {{z1'[t]}, {z2'[t]}, {z3'[t]}, {z4'[t]}} ==
     A.{{z1[t]}, {z2[t]}, {z3[t]}, {z4[t]}} + B.v;
   zEQ2 = {{z1[0]}, {z2[0]}, {z3[0]}, {z4[0]}} == {{0}, {0}, {0}, {0}};
   (* zEQ2 = {{z1[0]},{z2[0]}}== -Inverse[P... *)
   zs = (NDSolve[{zEQ1, zEQ2}, {z1[t], z2[t], {z3[t]}, {z4[t]}}, {t, 0, T}])[[
     1]];
   Return[({{z1[t]}, {z2[t]}, {z3[t]}, {z4[t]}} /. zs)];
  ];

(* Projection of xibar onto feasible space *)
Proj[xibar_, K_] :=
  Module[{xbar = xibar[[1]], ubar = xibar[[2]], xEQ1, xEQ2, xs},
   xEQ1 = {dX[[3]], dX[[4]]} == {(f[X, U] /. {u1[t] → (ubar + K.(X - xbar))[[1, 1]],
          u2[t] → (ubar + K.(X - xbar))[[2, 1]]})[[3]],
      (f[X, U] /. {u1[t] → (ubar + K.(X - xbar))[[1, 1]],
```

```
          u2[t] → (ubar + K.(X - xbar))[[2, 1]]}) [[4]]};
    xEQ2 = {{θ1[0]}, {θ2[0]}, {θ1'[0]}, {θ2'[0]}} == xi0[[1]];
    xs =
     (NDSolve[{xEQ1, xEQ2}, {θ1[t], θ2[t], θ1'[t], θ2'[t]}, {t, 0, T}])[[1]];
    Return[(X /. xs)];
   ];

combineInterps[interp_, maxIndex_, stepSize_] :=
  Module[{samples, index, val, retInterp}, samples = {};
   index = 0;
   While[index ≤ maxIndex, AppendTo[samples, {index, interp /. t → index}];
    index += stepSize;];
   Return[{Interpolation[samples, Method → "Hermite"][t]}];];

(* Armijo Line Search *)
Armijo[xi_, xibar_, zeta_, K_, maxIters_: 10] :=
  Module[{α = .01, β = .5, n = 0, xibarn, γ, X, U, Xn, Un, Jtemp, DJtemp},
   X = xi[[1]];
   U = xi[[2]];
   γ = β^n;
   xibarn = xibar + γ zeta;
   Xn  = Proj[xibarn, K];
   Un = xibarn[[2]] + K.(Xn - xibarn[[1]]);
   Un = {combineInterps[Un[[1, 1]], T, 1/100],
     combineInterps[Un[[2, 1]], T, 1/100]};
   Jtemp = J[X, U]; (* Only changes in main loop *)
   DJtemp = DJzeta[xi, zeta]; (* Only changes in main loop *)
   While[
    And[(J[Xn, Un])[[1, 1]] > (Jtemp + α γ DJtemp )[[1, 1]], n < maxIters],
    n = n + 1;
    γ = β^n;
    xibarn = xibar + γ zeta;
    Xn = Proj[xibarn, K];
    Un = xibarn[[2]] + K.(Xn - xibarn[[1]]);
    Un = {combineInterps[Un[[1, 1]], T, 1/100],
      combineInterps[Un[[2, 1]], T, 1/100]};
    Print["γ: ", γ];
   ];
   Return[{xibarn, {Xn, Un}}];
  ];

ϵ = 10^-2;
i = 0;
```

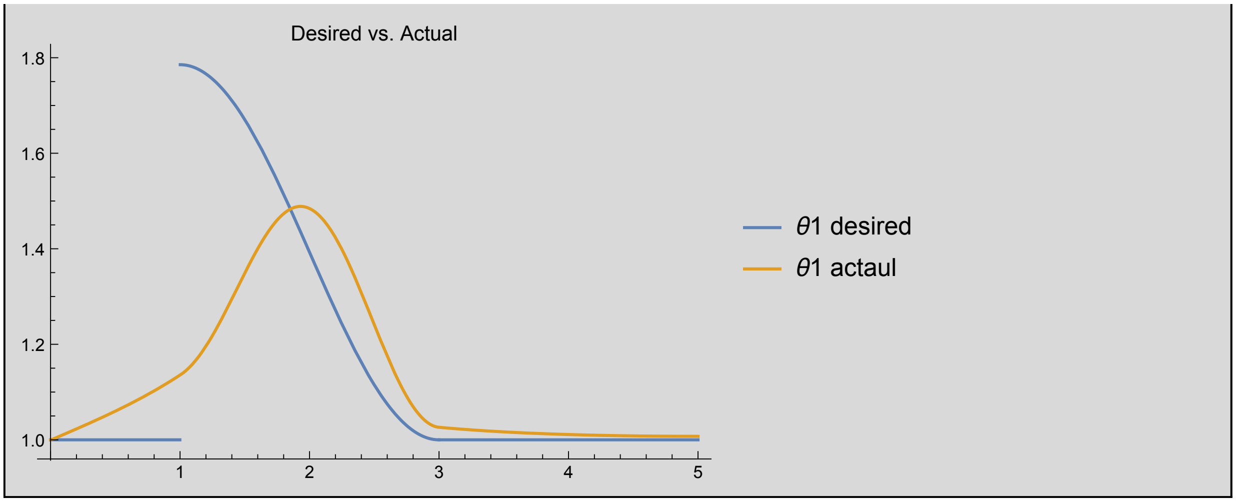
```
normi = 100;

(* Full Algorithm *)
While[
  And[Abs[normi] > ϵ, i < 30],
  A = Ta[Asym] /.
    {θ1[t] → xii[[1, 1, 1]], θ2[t] → xii[[1, 2, 1]], θ1'[t] → xii[[1, 3, 1]],
     θ2'[t] → xii[[1, 4, 1]], u1[t] → xii[[2, 1, 1]], u2[t] → xii[[2, 2, 1]]};
  A = {Flatten[{combineInterps[A[[1, 1]], T, 1/100],
      combineInterps[A[[1, 2]], T, 1/100], combineInterps[A[[1, 3]], T, 1/100],
      combineInterps[A[[1, 4]], T, 1/100]}],
    Flatten[{combineInterps[A[[2, 1]], T, 1/100],
      combineInterps[A[[2, 2]], T, 1/100], combineInterps[A[[2, 3]], T, 1/100],
      combineInterps[A[[2, 4]], T, 1/100]}],
    Flatten[{combineInterps[A[[3, 1]], T, 1/100],
      combineInterps[A[[3, 2]], T, 1/100], combineInterps[A[[3, 3]], T, 1/100],
      combineInterps[A[[3, 4]], T, 1/100]}],
    Flatten[{combineInterps[A[[4, 1]], T, 1/100],
      combineInterps[A[[4, 2]], T, 1/100], combineInterps[A[[4, 3]], T, 1/100],
      combineInterps[A[[4, 4]], T, 1/100]}]};
  B = Ta[Bsym] /. {θ2[t] → xii[[1, 2, 1]]};
  B = {Flatten[{combineInterps[B[[1, 1]], T, 1/100],
      combineInterps[B[[1, 2]], T, 1/100]}],
    Flatten[{combineInterps[B[[2, 1]], T, 1/100], combineInterps[
       B[[2, 2]], T, 1/100]}], Flatten[{combineInterps[B[[3, 1]], T, 1/100],
      combineInterps[B[[3, 2]], T, 1/100]}], Flatten[{combineInterps[
       B[[4, 1]], T, 1/100], combineInterps[B[[4, 2]], T, 1/100]}]};
  a = asym /. {θ1[t] → xii[[1, 1, 1]], θ2[t] → xii[[1, 2, 1]],
     θ1'[t] → xii[[1, 3, 1]], θ2'[t] → xii[[1, 4, 1]]};
  b = bsym /. {u1[t] → xii[[2, 1, 1]], u2[t] → xii[[2, 2, 1]]};
  Pn = Psol[A, B, Qn, Rn, P1n];
  rn = rsol[A, B, a, b, Pn, Rn, P1n, xii];
  z = zsol[A, B, b, Pn, rn];
  v = -Inverse[Rn].(b + B^T.Pn.z + B^T.rn);
  v =
   {combineInterps[v[[1, 1]], T, 1/100], combineInterps[v[[2, 1]], T, 1/100]};
  zeta = {z, v};
  Pr = Psol[A, B, Qr, Rr, P1r];
  κ = -Inverse[Rr].B^T.Pr;
  κ = {Flatten[{combineInterps[κ[[1, 1]], T, 1/100],
      combineInterps[κ[[1, 2]], T, 1/100], combineInterps[
       κ[[1, 3]], T, 1/100], combineInterps[κ[[1, 4]], T, 1/100]}],
    Flatten[{combineInterps[κ[[2, 1]], T, 1/100], combineInterps[
```

```
      κ[[2, 2]], T, 1/100], combineInterps[κ[[2, 3]], T, 1/100],
     combineInterps[κ[[2, 4]], T, 1/100]}]};
  {xibar_{i+1}, xi_{i+1}} = Armijo[xi_i, xibar_i, zeta, κ];
  xibar_{i+1} = {{combineInterps[xibar_{i+1}[[1, 1, 1]], T, 1/100],
     combineInterps[xibar_{i+1}[[1, 2, 1]], T, 1/100],
     combineInterps[xibar_{i+1}[[1, 3, 1]], T, 1/100],
     combineInterps[xibar_{i+1}[[1, 4, 1]], T, 1/100]},
    {combineInterps[xibar_{i+1}[[2, 1, 1]], T, 1/100],
     combineInterps[xibar_{i+1}[[2, 2, 1]], T, 1/100]}};
  xi_{i+1} = {{combineInterps[xi_{i+1}[[1, 1, 1]], T, 1/100],
     combineInterps[xi_{i+1}[[1, 2, 1]], T, 1/100], combineInterps[
      xi_{i+1}[[1, 3, 1]], T, 1/100], combineInterps[xi_{i+1}[[1, 4, 1]], T, 1/100]},
    {combineInterps[xi_{i+1}[[2, 1, 1]], T, 1/100],
     combineInterps[xi_{i+1}[[2, 2, 1]], T, 1/100]}};
  norm_{i+1} = (DJzeta[xi_i, zeta])[[1, 1]];
  Print[xi_{i+1}[[1]] /. t → T];
  Print[norm_{i+1}];
  i = i + 1;
  Clear[A, B, a, b, Pn, rn, z, v, zeta, Pr, κ]
 ];

Print["Number of iterations: ", i];
(* Plot the trajectories and control effort *)
Plot[{θ1d[t], xi_i[[1, 1, 1]]}, {t, 0, T}, PlotRange → Full ,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"θ1 desired", "θ1 actaul"}]
Plot[{θ2d[t], xi_i[[1, 2, 1]]}, {t, 0, T}, PlotRange → Full ,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"θ2 desired", "θ2 actaul"}]
Plot[{dθ1d[t], xi_i[[1, 3, 1]]}, {t, 0, T},
 PlotRange → Full , PlotLabel → "Desired vs. Actual",
 PlotLegends → {"θ1' desired", "θ1' actaul"}]
Plot[{dθ2d[t], xi_i[[1, 4, 1]]}, {t, 0, T},
 PlotRange → Full , PlotLabel → "Desired vs. Actual",
 PlotLegends → {"θ2' desired", "θ2' actaul"}]
Plot[{u1d[t], xi_i[[2, 1, 1]]}, {t, 0, T}, PlotRange → Full,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"u1 desired ", "u1 actual"}]
Plot[{u2d[t], xi_i[[2, 2, 1]]}, {t, 0, T}, PlotRange → Full,
 PlotLabel → "Desired vs. Actual", PlotLegends → {"u2 desired ", "u2 actual"}]
ListLinePlot[Table[{h, Abs[norm_h]}, {h, 1, i}], Filling → Axis]

(*ANIMATION*)
(*x and y coordinates for pendulum 1 *)
X1[τ_] := l11 * Sin[xi_i[[1, 1, 1]]] /. t → τ;
Y1[τ_] := -l11 * Cos[xi_i[[1, 1, 1]]] /. t → τ;
```

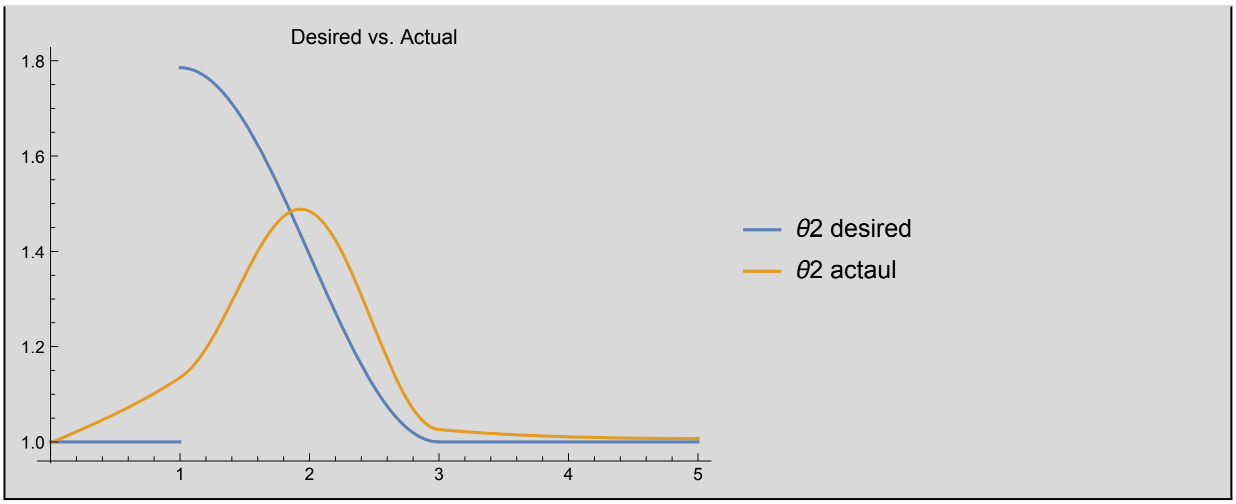
```
(*x and y coordinates for pendulum 2*)
X2[τ_] :=
  ll1 * Sin[xiᵢ[[1, 1, 1]]] + ll2 * Sin[xiᵢ[[1, 1, 1]] + xiᵢ[[1, 2, 1]]] /. t → τ;
Y2[τ_] := -ll1 * Cos[xiᵢ[[1, 1, 1]]] -
    ll2 * Cos[xiᵢ[[1, 1, 1]] + xiᵢ[[1, 2, 1]]] /. t → τ;

Animate[Show [Graphics[{Purple, Thick, Line[{{0, 0}, {X1[τ], Y1[τ]}}],
    Purple, Thick, Line[{{X1[τ], Y1[τ]}, {X2[τ], Y2[τ]}}],
    Black, Line[{{-3, 0}, {3, 0}}], Line[{{-3, 0}, {-3, -2}}]}],
  AspectRatio → Automatic, PlotRange → {{-2.1, 2.1}, {-2.1, 2.1}},
  Frame → True], {τ, 0, 2}, AnimationRate → 1]

(*The trajectory plot of the Double Pendulum*)
ParametricPlot[{{X1[τ], Y1[τ]}, {X2[τ], Y2[τ]}},
 {τ, 0, T}, AspectRatio → Automatic, AxesLabel → {x, y},
 PlotLegends → {"Trajectory of pendulum 1", "Trajectory of pendulum 2"}]
```

```
γ: 0.5
{{0.99745}, {1.02728}, {-0.0340022}, {0.0438479}}
-12.7226
{{1.00804}, {1.00701}, {-0.0239203}, {-0.00626057}}
-11.7835
{{1.00692}, {1.00659}, {-0.019331}, {-0.00294147}}
-0.0807045
{{1.00737}, {1.00687}, {-0.0182736}, {-0.00350336}}
-0.0251303
{{1.00734}, {1.00687}, {-0.0171499}, {-0.00326005}}
-0.00407125
Number of iterations: 5
```

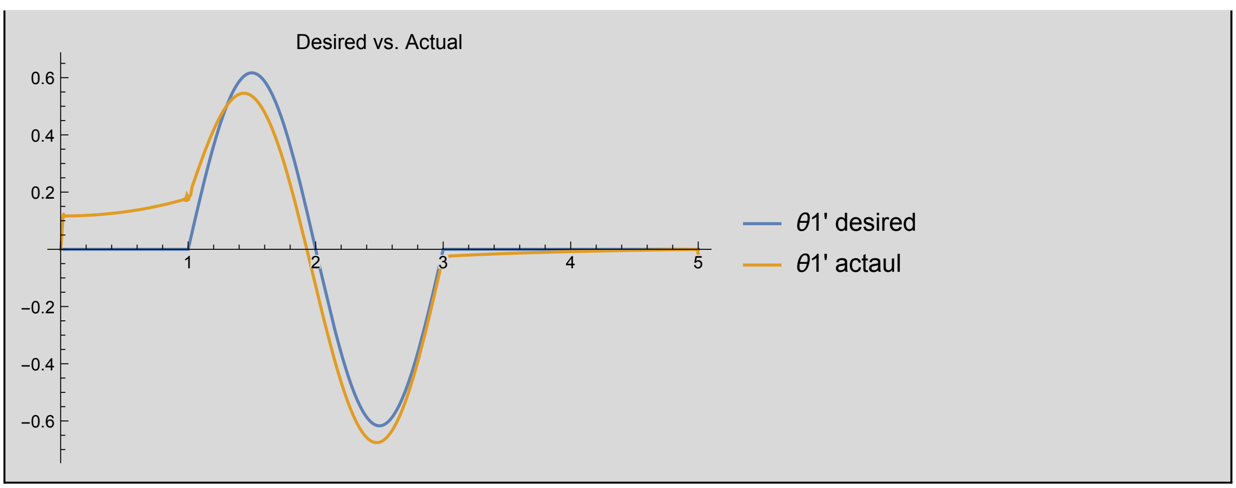
Out[220]=



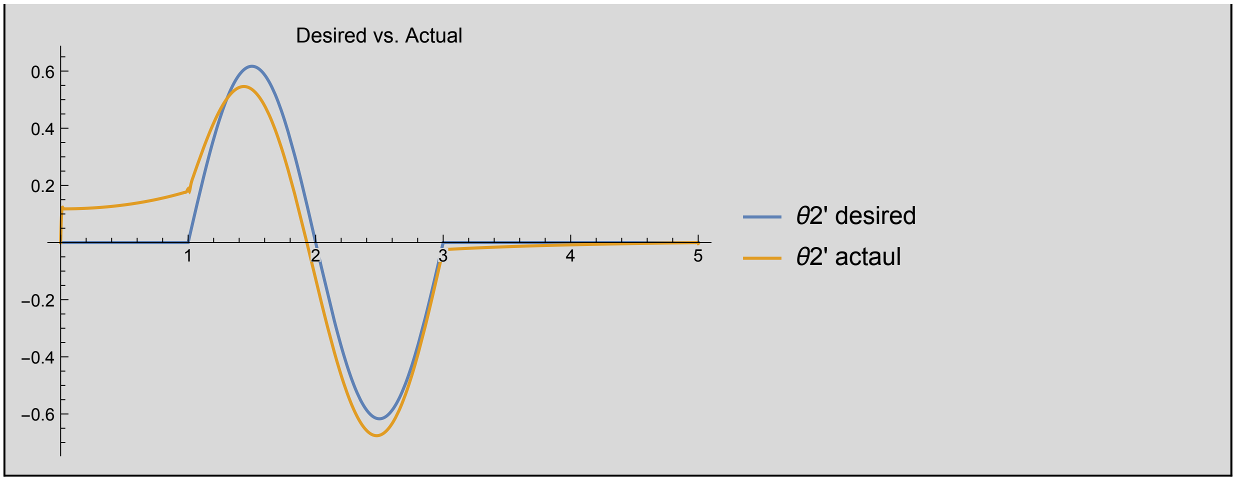
Out[221]=



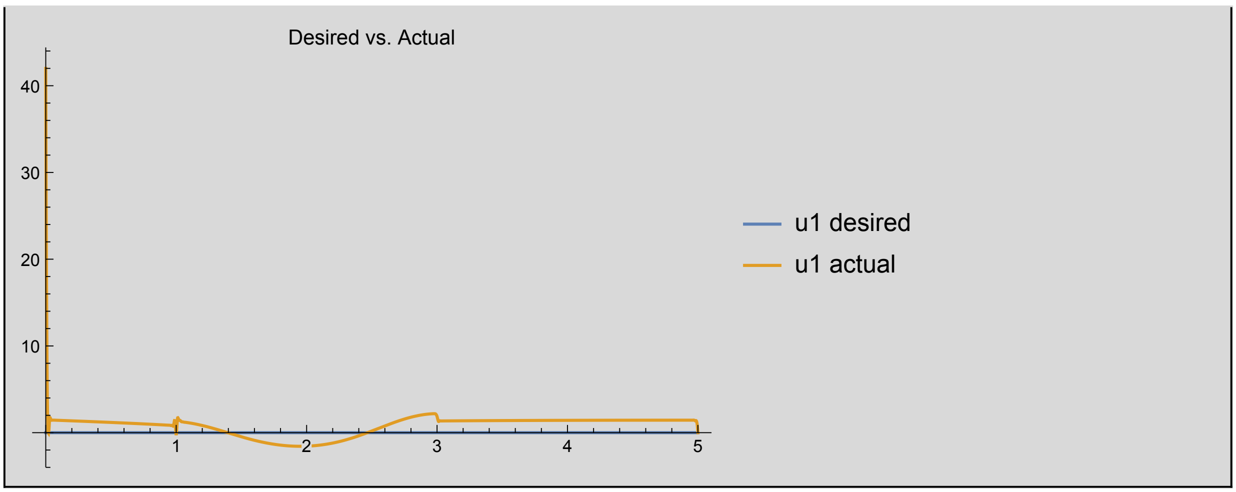
Out[222]=



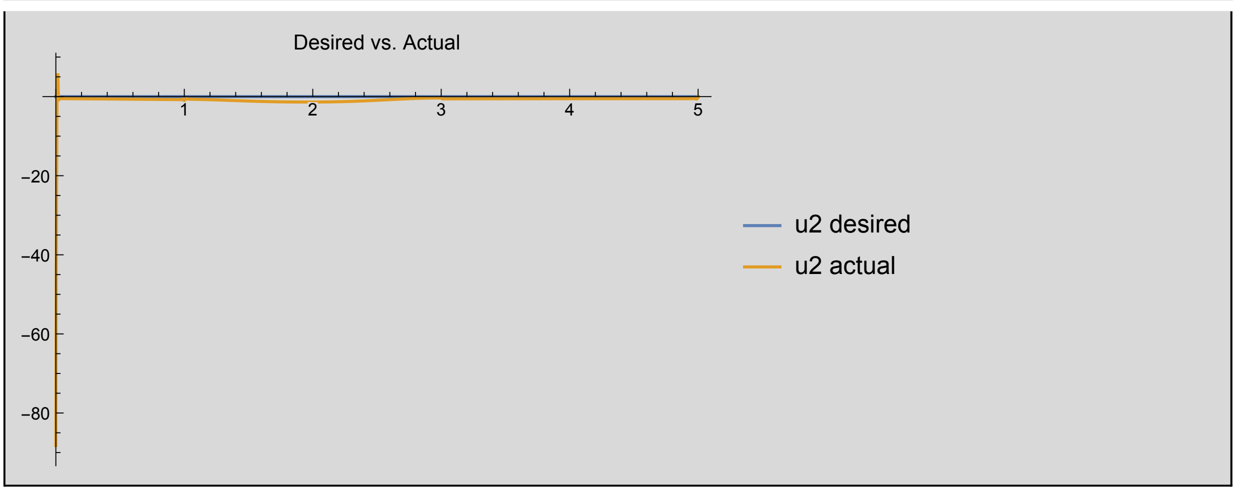
Out[223]=



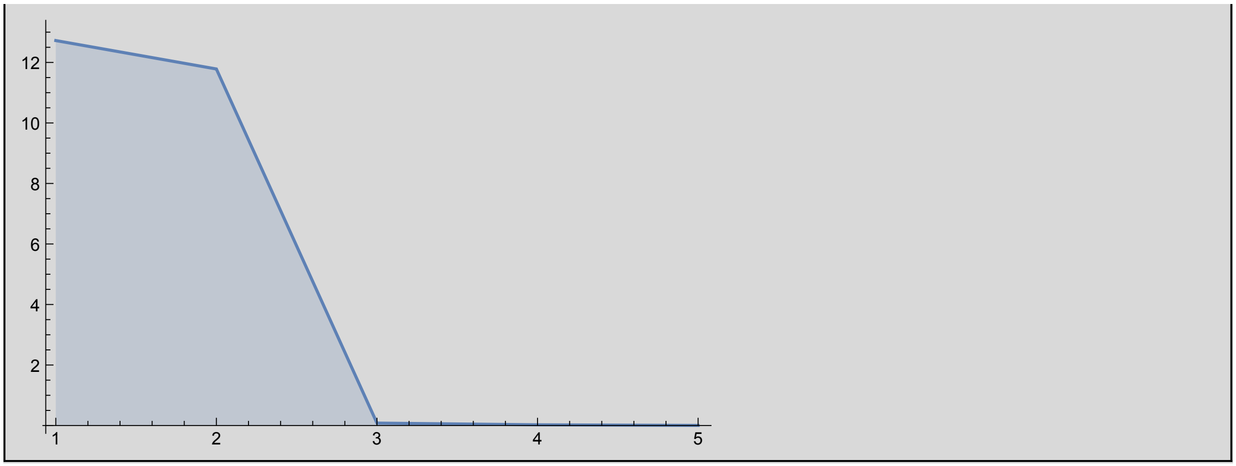
Out[224]=



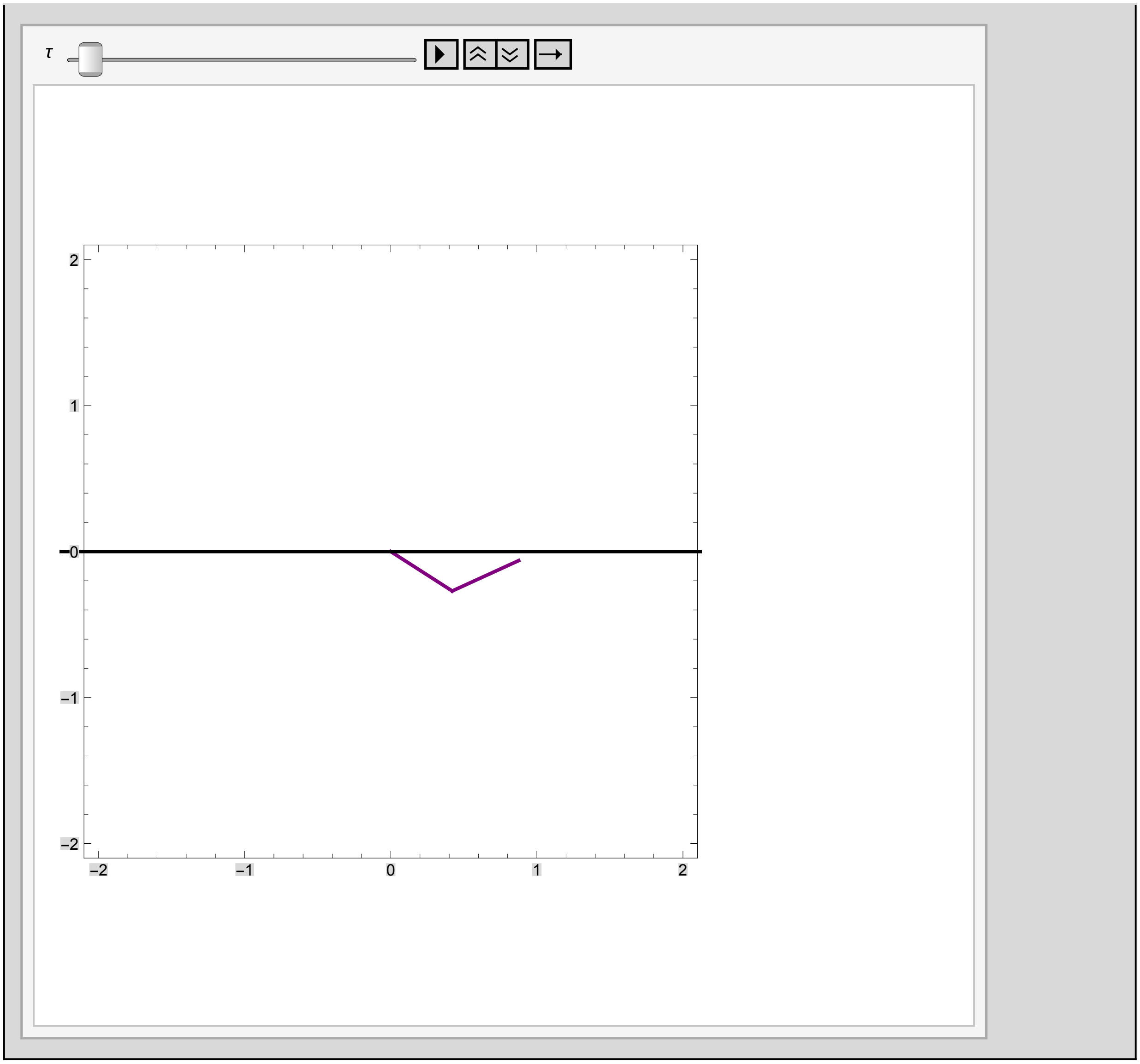
Out[225]=



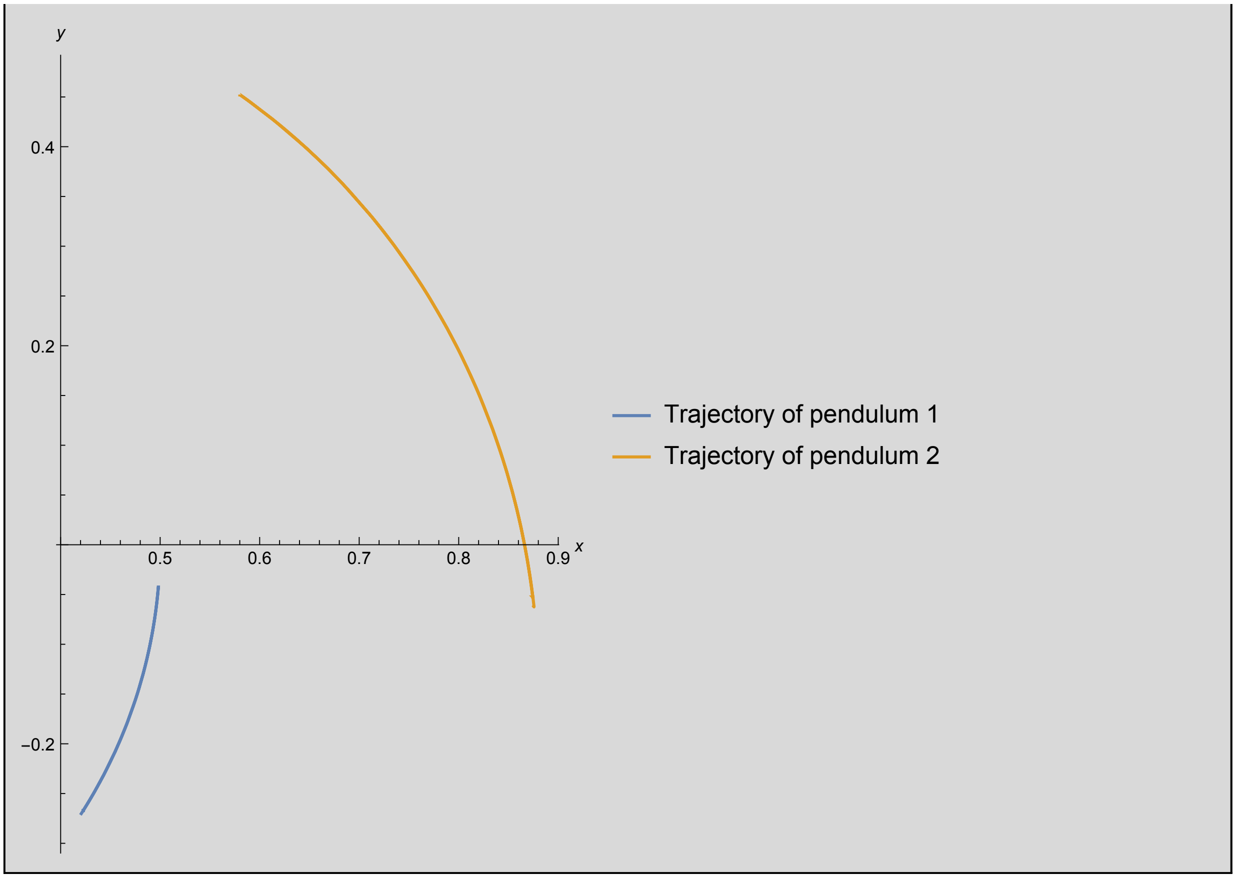
Out[226]=



Out[231]=



Out[232]=


y
0.4
0.2
−0.2
0.5
0.6
0.7
0.8
0.9
x
Trajectory of pendulum 1
Trajectory of pendulum 2