Computing the Adjoint Equations given a cost function symbolically

In[1211]:=
```
Clear["Global`*"]; (* Fix Bugs here *)
Remove["Global`*"];

Clear[u];
n = 20; τ = 15; τ1 = τ * 1.25 ;
nDim = 4;
Δt = τ/n;
xState = {x, xdot, θ, θdot};
λ = {λ1, λ2, λ3, λ4};

fx =
  {x2, (A * Sin[θ1] * Cos[θ1] + A * Sin[θ1] * θ2^2 + u) / (1 - A * Cos[θ1]^2), θ2, -Sin[θ1] -
     Cos[θ1] * (A * Sin[θ1] * Cos[θ1] + A * Sin[θ1] * θ2^2 + u) / (1 - A * Cos[θ1]^2)} /.
   {x1 → x, x2 → xdot, θ1 → θ, θ2 → θdot};

L = 1 / 2 * u^2 + 1 / 2 * Weight * (xState - {0, 0, π, 0}).(xState - {0, 0, π, 0});
(*Cost function*)
eqn = D[fx, u].λ + D[L, u];
sol = Solve[{eqn == 0}, {u}][[1]];
{u} = {u} /. sol ;
λdot = -Grad[fx, xState]ᵀ.λ - Grad[L, xState]ᵀ;
fSymbolic = Simplify[Join[fx, λdot]]
Dimensions[fSymbolic]
```

Out[1224]=

$$\Bigg\{\text{xdot},\ \frac{\frac{\lambda2-\lambda4\,\text{Cos}[\theta]}{-1+\text{A}\,\text{Cos}[\theta]^2}+\text{A}\,\theta\text{dot}^2\,\text{Sin}[\theta]+\text{A}\,\text{Cos}[\theta]\,\text{Sin}[\theta]}{1-\text{A}\,\text{Cos}[\theta]^2},\ \theta\text{dot},$$

$$-\text{Sin}[\theta]-\frac{\text{Cos}[\theta]\left(\frac{\lambda2-\lambda4\,\text{Cos}[\theta]}{-1+\text{A}\,\text{Cos}[\theta]^2}+\text{A}\,\theta\text{dot}^2\,\text{Sin}[\theta]+\text{A}\,\text{Cos}[\theta]\,\text{Sin}[\theta]\right)}{1-\text{A}\,\text{Cos}[\theta]^2},\ -\text{Weight}\,\text{x},$$

$$-\text{Weight}\,\text{xdot}-\lambda1,\ \frac{1}{\left(-1+\text{A}\,\text{Cos}[\theta]^2\right)^3}\Big(-\pi\,\text{Weight}+\text{Weight}\,\theta+\text{A}^2\left(\text{A}\,\theta\text{dot}^2\,\lambda2-\lambda4\right)\text{Cos}[\theta]^5+$$

$$\text{A}^3\left(\pi\,\text{Weight}-\text{Weight}\,\theta+\lambda2-\theta\text{dot}^2\,\lambda4\right)\text{Cos}[\theta]^6-\text{A}\left(\lambda2-\theta\text{dot}^2\,\lambda4\right)\text{Sin}[\theta]^2+$$

$$\text{A}\,\text{Cos}[\theta]^2\left(3\,\pi\,\text{Weight}-3\,\text{Weight}\,\theta+\lambda2-\theta\text{dot}^2\,\lambda4-3\,\lambda2\,\lambda4\,\text{Sin}[\theta]\right)+$$

$$\text{Cos}[\theta]\left(\text{A}\,\theta\text{dot}^2\,\lambda2-\lambda4+\lambda4^2\,\text{Sin}[\theta]-2\,\text{A}^2\,\theta\text{dot}^2\,\lambda2\,\text{Sin}[\theta]^2\right)+$$

$$\text{A}\,\text{Cos}[\theta]^3\left(-2\,\text{A}\,\theta\text{dot}^2\,\lambda2+2\,\lambda4+\lambda4^2\,\text{Sin}[\theta]+2\,\text{A}\left(\text{A}\,\theta\text{dot}^2\,\lambda2-\lambda4\right)\text{Sin}[\theta]^2\right)+$$

$$\text{A}^2\,\text{Cos}[\theta]^4\left(-3\,\pi\,\text{Weight}+3\,\text{Weight}\,\theta-2\,\lambda2+2\,\theta\text{dot}^2\,\lambda4+\text{A}\left(\lambda2-\theta\text{dot}^2\,\lambda4\right)\text{Sin}[\theta]^2\right)+$$

$$\text{A}\,\lambda2^2\,\text{Sin}[2\,\theta]+\lambda4\,\text{Sin}[\theta]\,(-\lambda2+\text{A}\,\text{Sin}[2\,\theta])\Big),$$

$$\frac{\text{Weight}\,\theta\text{dot}+\lambda3-\text{A}\,(\text{Weight}\,\theta\text{dot}+\lambda3)\,\text{Cos}[\theta]^2+2\,\text{A}\,\theta\text{dot}\,\lambda2\,\text{Sin}[\theta]-2\,\text{A}\,\theta\text{dot}\,\lambda4\,\text{Cos}[\theta]\,\text{Sin}[\theta]}{-1+\text{A}\,\text{Cos}[\theta]^2}\Bigg\}$$

Out[1225]= {8}

In[904]:= **Lnew**

Out[904]= $$\frac{1}{2}\left(x^2+xdot^2+(-\pi+\theta)^2+\theta dot^2\right)+\frac{(\lambda 2-\lambda 4\,\mathrm{Cos}[\theta])^2}{2\left(-1+A\,\mathrm{Cos}[\theta]^2\right)^2}$$

In[1226]:=
```
ClearAll["Global`*"];
Remove["Global`*"];
(*ICs - Initial Conditions *) (* Fix Feedback *)
ffCartPendulum[ICs_,n_,τ_,τ1_,A_,order_,maxIter_,InitGuess_,Weight_]:=
Module[{x,dist,xdot,f,θ,θdot,λ1,λ2,λ3,λ4,Δt,bcs,eqns,sv,froot,xff,xdotff,xff0,xdotff0,θff0,θdotf

f[{x_,xdot_,θ_,θdot_,λ1_,λ2_,λ3_,λ4_}]:=
{xdot, ((λ2-λ4 Cos[θ])/(-1+A Cos[θ]^2) +A θdot^2 Sin[θ]+A Cos[θ] Sin[θ])/(1-A Cos[θ]^2) ,θdot,-Sin[θ]- (Cos[θ] ((λ2-λ4 Cos[θ])/(-1+A Cos[θ]^2) +A θdot^2 Sin[θ]+A
                                                                                                   /(1-A Cos[θ]^2)

xGuess = Table[If[i ≠ -1,xGuess_(i+1) = xGuess_i +Δt*f[xGuess_i],xGuess_0 = {ICs[[1]],ICs[[2]],ICs[[3]],ICs[[4
 bcs={Subscript[x, 0]==ICs[[1]],Subscript[xdot, 0]==ICs[[2]],Subscript[x, n]==Subscript[xdot, n]==0,Sub
eqns=Flatten[Join[bcs,Table[Thread[{Subscript[x, i],Subscript[xdot, i],Subscript[θ, i],Subscript
1/2 Δt (f[{Subscript[x, i-1],Subscript[xdot, i-1],Subscript[θ, i-1],Subscript[θdot, i-1],Subscri
f[{Subscript[x, i],Subscript[xdot, i],Subscript[θ, i],Subscript[θdot, i],Subscript[λ1, i],Subscr
{Subscript[x, i-1],Subscript[xdot, i-1],Subscript[θ, i-1],Subscript[θdot, i-1],Subscript[λ1, i-1
sv = Flatten[Table[{{Subscript[x, i],xGuess[[i+1]][[1]]},{Subscript[xdot, i],xGuess[[i+1]][[2]]},{Subscr
                    {Subscript[λ1, i],xGuess[[i+1]][[5]]},{Subscript[λ2, i],xGuess[[i+1]][[6]]},{Subscr
froot=FindRoot[eqns,sv,MaxIterations→maxIter];
xff0=ListInterpolation[Table[Subscript[x, i],{i,0,n}]/. froot,{0,τ},InterpolationOrder→order];
xdotff0=ListInterpolation[Table[Subscript[xdot, i],{i,0,n}]/. froot,{0,τ},InterpolationOrder→ord
θff0=ListInterpolation[Table[Subscript[θ, i],{i,0,n}]/. froot,{0,τ},InterpolationOrder→order];
θdotff0=ListInterpolation[Table[Subscript[θdot, i],{i,0,n}]/. froot,{0,τ},InterpolationOrder→ord
λ1ff0 = ListInterpolation[Table[Subscript[λ1, i],{i,0,n}]/. froot,{0,τ},InterpolationOrder→order
λ2ff0 = ListInterpolation[Table[Subscript[λ2, i],{i,0,n}]/. froot,{0,τ},InterpolationOrder→order
λ3ff0 = ListInterpolation[Table[Subscript[λ3, i],{i,0,n}]/. froot,{0,τ},InterpolationOrder→order
λ4ff0 = ListInterpolation[Table[Subscript[λ3, i],{i,0,n}]/. froot,{0,τ},InterpolationOrder→order

uff0=ListInterpolation[Table[1/(1-A Cos[Subscript[θ, i]]^2) (Subscript[λ4, i]Cos[Subscript[θ, i]

xff[t_]:=Piecewise[{{xff0[t],0≤t≤τ}},0];
xdotff[t_]:=Piecewise[{{xdotff0[t],0≤t≤τ}},0];
θff[t_]:=Piecewise[{{θff0[t],0≤t≤τ}},π];
θdotff[t_]:=Piecewise[{{θdotff0[t],0≤t≤τ}},0];
uff[t_]:=Piecewise[{{uff0[t],0≤t≤τ}},0];

{xff,xdotff,θff,θdotff,uff,λ1ff0,λ2ff0,λ3ff0,λ4ff0}]

testSwingUp[ICs_,τ_,τ1_,uff0_,A_]:=Module[{eq,init,x,xdot,θ,θdot,xs,xdots,θs,θdots,t,J},
eq={x'[t]==xdot[t],xdot'[t]==1/(1-A Cos[θ[t]]^2) (uff0[t]+A θdot[t]^2 Sin[θ[t]]+A Cos[θ[t]] Sin
init={x[0]==ICs[[1]],xdot[0]==ICs[[2]],θ[0]==ICs[[3]],θdot[0]==ICs[[4]]};
{xs,xdots,θs,θdots}=NDSolveValue[{eq,init},{x,xdot,θ,θdot},{t,0,τ1},Method→{"DiscontinuityProces
J = NIntegrate[uff0[t]^2,{t,0,τ}];
```

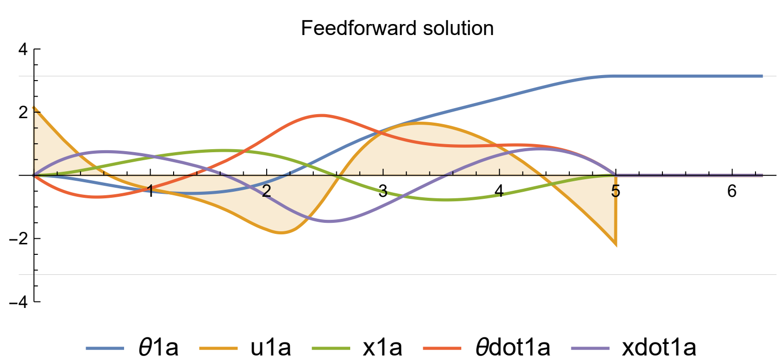
```
{xs,xdots,θs,θdots,uff0,J}]


CalculateSMatrix[x1a_,xdot1a_,θ1a_,θdot1a_,u1a_,τ_,A_]:= Module[{x,L,RHS,xdot,θ,θdot,u,K,S,soltn

xState = {x,xdot,θ,θdot};
x2dot = 1/(1-A Cos[θ]^2) (u+A θdot^2 Sin[θ]+A Cos[θ] Sin[θ]);
θ2dot= 1/(1-A Cos[θ]^2) (-Sin[θ]-Cos[θ] (u+A θdot^2 Sin[θ]));
fx = {xdot,x2dot,θdot,θ2dot};
L = 1/2*u^2 + 1/2*(xState - {0,0,π,0}).(xState - {0,0,π,0}); (*Cost function*)
Af = Grad[fx,xState]; (* For nD stuff use Grad*)
Bf = D[fx,u] ;(*For 1D stuff use D*)
Q = Grad[Grad[L,xState],xState]; (* Fix this *)
Mf = Grad[D[L,u],xState];
R = D[L,{u,2}];
S0 = (0 0 0 0
      0 0 0 0
      0 0 0 0
      0 0 0 0);
RHS[t_] := (IdentityMatrix[4] + Af^T.S[t] + S[t].Af - KroneckerProduct[S[t].Bf,Bf^T.S[t]]) /. {x→
sol2 = S /. NDSolve[{S'[t]== RHS[t],S[0]==S0},S,{t,0,τ }];
S = sol2[[1]]
]
CalculateGains[x1a_,xdot1a_,θ1a_,θdot1a_,u1a_,time_,A_,τ_,S_]:= Module[{x,L,RHS,xdot,θ,θdot,u,K,
xState = {x,xdot,θ,θdot};
x2dot = 1/(1-A Cos[θ]^2) (u+A θdot^2 Sin[θ]+A Cos[θ] Sin[θ]);
θ2dot= 1/(1-A Cos[θ]^2) (-Sin[θ]-Cos[θ] (u+A θdot^2 Sin[θ]));
fx = {xdot,x2dot,θdot,θ2dot};
Bf = D[fx,u] ;(*For 1D stuff use D*)
K = (Bf^T.S[τ - time])/. {x→ x1a[time], xdot → xdot1a[time], θ →θ1a[time], θdot → θdot1a[time],
K
]
testWithFB[ICs_,τ_,τ1_,xff0_,xdotff0_,θff0_,θdotff0_,uff0_,A_]:=Module[{eq,init,θ,θdot,θff,θdot
κ1=κ2=3;  (* lqr for q=r for balancing pendulum *)
κ3 = -0.1;κ4 = -0.65;
xff[t_]:=Piecewise[{{xff0[t],0≤t≤τ}},0];
xdotff[t_]:=Piecewise[{{xdotff0[t],0≤t≤τ}},0];
θff[t_]:=Piecewise[{{θff0[t],0≤t≤τ}},π];
θdotff[t_]:=Piecewise[{{θdotff0[t],0≤t≤τ}},0];
uff[t_]:=Piecewise[{{uff0[t],0≤t≤τ}},0];
S = CalculateSMatrix[xff,xdotff,θff,θdotff,uff,τ,A];
K[t_] := CalculateGains[xff,xdotff,θff,θdotff,uff,t,A,τ,S];
ufb[t_] := Piecewise[{{
 K[t].{xff[t]-x[t],xdotff[t]-xdot[t],θff[t]-θ[t],θdotff[t]-θdot[t]},0≤t≤τ}},κ1(θff[t]-θ[t])+κ2

eq={x'[t]==xdot[t],xdot'[t]==1/(1-A Cos[θ[t]]^2) (u[t]+A θdot[t]^2 Sin[θ[t]]+A Cos[θ[t]] Sin[θ[

init={x[0]==ICs[[1]],xdot[0]==ICs[[2]],θ[0]==ICs[[3]],θdot[0]==ICs[[4]]};
```

```
{xs,xdots,θs,θdots}=NDSolveValue[{eq,init},{x,xdot,θ,θdot},{t,0,τ1},Method→{"DiscontinuityProces
us[t_]:=uff[t]+Piecewise[{{K[t].{xff[t]-xs[t],xdotff[t]-xdots[t],θff[t]-θs[t],θdotff[t]-θdots[
J = NIntegrate[us[t]^2,{t,0,τ}];
{xs,xdots,θs,θdots,us,J}]
```

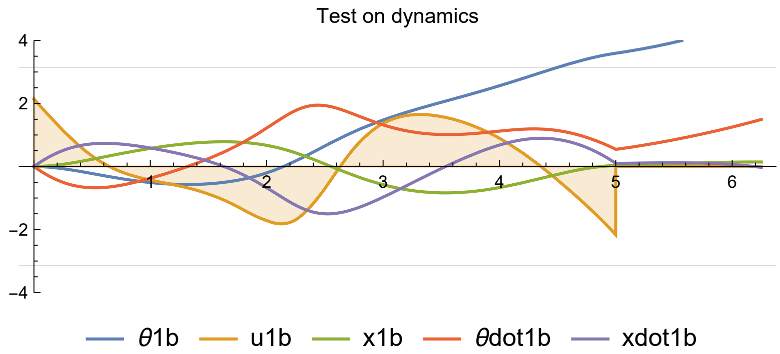
The idea is to use the cost L = 1/2*u^2 + 1/2*(δx)ᵀT(δx) where δx = x(t) - r(t), r(t) = {0,0,0π,0}. The advantage would be that now we can choose a fixed time τ to be very large and in case the solution can be obtained in a smaller time, it will be done as we are minimizing this cost.

In[1558]:=
```
n = 20; τ = 5; τ1 = τ * 1.25; A = 0.2; order = 4; maxIter = 100; Weight = 0;
ICs = {-0.048555026816794494`,
   -1.560065966757075`, 0.7613152376955525`, 1.9382391342292873`};
ICs = {0, 0, 0, 0};
InitGuess =
  {RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}]};
InitGuess = {0, 0, 0, 0};
{x1a, xdot1a, θ1a, θdot1a, u1a, λ1ff0, λ2ff0, λ3ff0, λ4ff0} =
  ffCartPendulum[ICs, n, τ, τ1, A, order, maxIter, InitGuess, Weight];
{x1b, xdot1b, θ1b, θdot1b, u1b, J1} = testSwingUp[ICs, τ, τ1, u1a, A];
{x1c, xdot1c, θ1c, θdot1c, u1c, J} =
 testWithFB[ICs, τ, τ1, x1a, xdot1a, θ1a, θdot1a, u1a, A];
p1a = Plot[{θ1a[t], u1a[t], x1a[t], θdot1a[t], xdot1a[t]}, {t, 0, τ1}, Filling → {2 → Axis},
  PlotRange → {-4, 4}, PlotLegends → {"θ1a", "u1a", "x1a", "θdot1a", "xdot1a"},
  PlotLabel → "Feedforward solution", AspectRatio → 1 / 3,
  ImageSize → 400, GridLines → {None, {-π, π}}]
p1b = Plot[{θ1b[t], u1a[t], x1b[t], θdot1b[t], xdot1b[t]},
  {t, 0, τ1}, PlotRange → {-4, 4}, Filling → {2 → Axis},
  PlotLegends → {"θ1b", "u1b", "x1b", "θdot1b", "xdot1b"}, PlotLabel → "Test on dynamics",
  AspectRatio → 1 / 3, ImageSize → 400, GridLines → {None, {-π, π}}]
p1c = Plot[{θ1c[t], u1c[t], x1c[t], θdot1c[t], xdot1c[t]}, {t, 0, τ1}, PlotRange → {-4, 4},
  Filling → {2 → Axis}, PlotLegends → {"θ1c", "u1c", "x1c", "θdot1c", "xdot1c"},
  PlotLabel → "LQR feedback solution", AspectRatio → 1 / 3,
  ImageSize → 400, GridLines → {None, {-π, π}}]
```

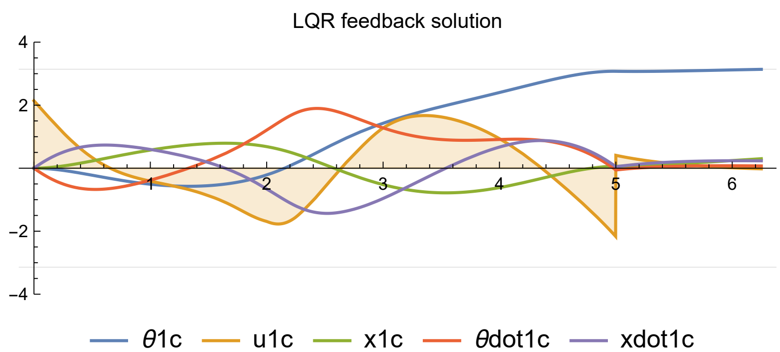
Out[1565]=



Out[1566]=



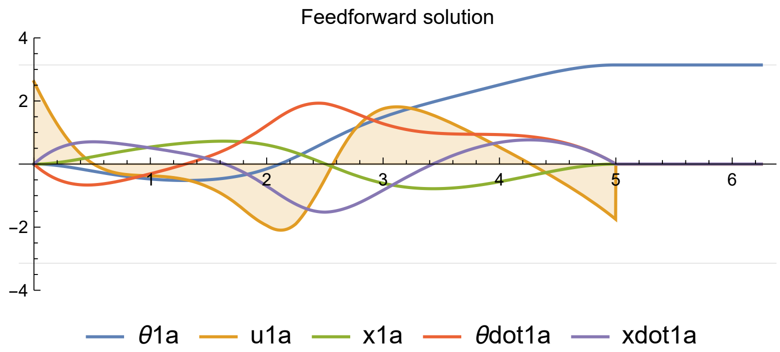
Out[1567]=



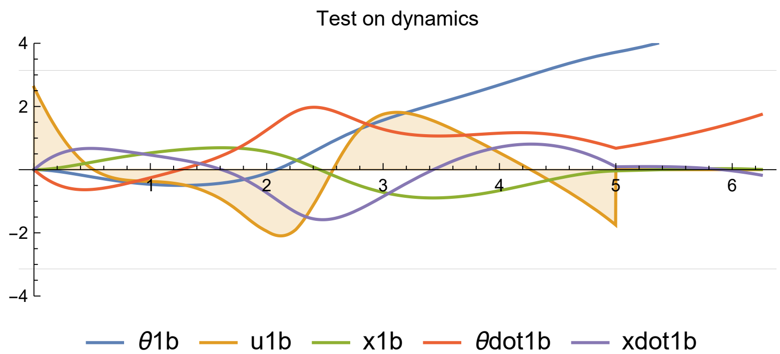
In[1667]:=

```
n = 20; τ = 5; τ1 = τ * 1.25; A = 0.2; order = 4; maxIter = 100; Weight = 0.4;
ICs = {-0.048555026816794494`,
   -1.560065966757075`, 0.7613152376955525`, 1.9382391342292873`};
ICs = {0, 0, 0, 0};
InitGuess =
  {RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}]};
InitGuess = {0, 0, 0, 0};
{x1a, xdot1a, θ1a, θdot1a, u1a, λ1ff0, λ2ff0, λ3ff0, λ4ff0} =
  ffCartPendulum[ICs, n, τ, τ1, A, order, maxIter, InitGuess, Weight];
{x1b, xdot1b, θ1b, θdot1b, u1b, J1} = testSwingUp[ICs, τ, τ1, u1a, A];
{x1c, xdot1c, θ1c, θdot1c, u1c, J} =
 testWithFB[ICs, τ, τ1, x1a, xdot1a, θ1a, θdot1a, u1a, A];
p1a = Plot[{θ1a[t], u1a[t], x1a[t], θdot1a[t], xdot1a[t]}, {t, 0, τ1}, Filling → {2 → Axis},
  PlotRange → {-4, 4}, PlotLegends → {"θ1a", "u1a", "x1a", "θdot1a", "xdot1a"},
  PlotLabel → "Feedforward solution", AspectRatio → 1 / 3,
  ImageSize → 400, GridLines → {None, {-π, π}}]
p1b = Plot[{θ1b[t], u1a[t], x1b[t], θdot1b[t], xdot1b[t]},
  {t, 0, τ1}, PlotRange → {-4, 4}, Filling → {2 → Axis},
  PlotLegends → {"θ1b", "u1b", "x1b", "θdot1b", "xdot1b"}, PlotLabel → "Test on dynamics",
  AspectRatio → 1 / 3, ImageSize → 400, GridLines → {None, {-π, π}}]
p1c = Plot[{θ1c[t], u1c[t], x1c[t], θdot1c[t], xdot1c[t]}, {t, 0, τ1}, PlotRange → {-4, 4},
  Filling → {2 → Axis}, PlotLegends → {"θ1c", "u1c", "x1c", "θdot1c", "xdot1c"},
  PlotLabel → "LQR feedback solution", AspectRatio → 1 / 3,
  ImageSize → 400, GridLines → {None, {-π, π}}]
```

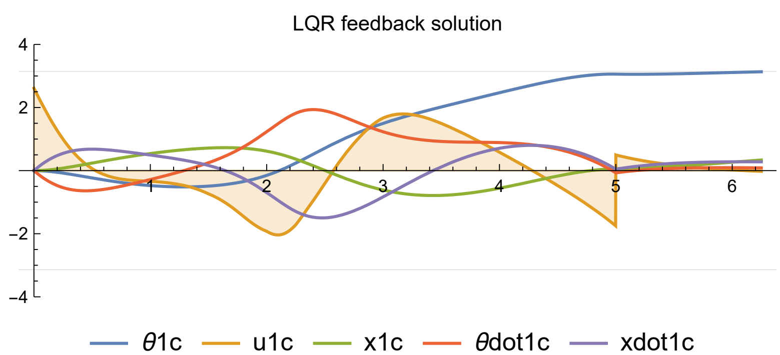
Out[1674]=



Out[1675]=



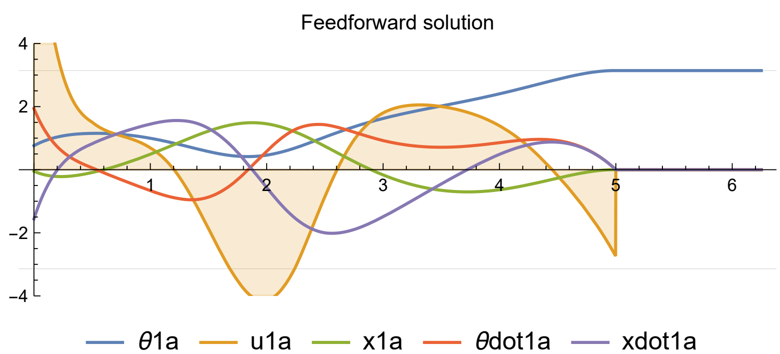
Out[1676]=



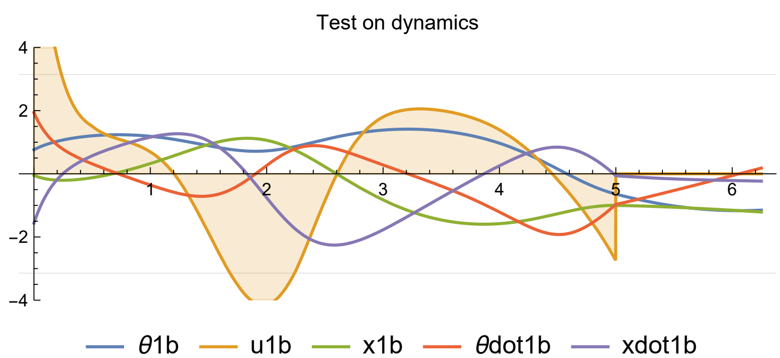
Non zero Initial Conditions worked but required repeated re initializations

In[1710]:=
```
n = 20; τ = 5; τ1 = τ * 1.25; A = 0.2; order = 4; maxIter = 100; Weight = 0.4;
ICs = {-0.048555026816794494`,
   -1.560065966757075`, 0.7613152376955525`, 1.9382391342292873`};
InitGuess =
  {RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}]};

{x1a, xdot1a, θ1a, θdot1a, u1a, λ1ff0, λ2ff0, λ3ff0, λ4ff0} =
  ffCartPendulum[ICs, n, τ, τ1, A, order, maxIter, InitGuess, Weight];
{x1b, xdot1b, θ1b, θdot1b, u1b, J1} = testSwingUp[ICs, τ, τ1, u1a, A];
{x1c, xdot1c, θ1c, θdot1c, u1c, J} =
 testWithFB[ICs, τ, τ1, x1a, xdot1a, θ1a, θdot1a, u1a, A];
p1a = Plot[{θ1a[t], u1a[t], x1a[t], θdot1a[t], xdot1a[t]}, {t, 0, τ1}, Filling → {2 → Axis},
  PlotRange → {-4, 4}, PlotLegends → {"θ1a", "u1a", "x1a", "θdot1a", "xdot1a"},
  PlotLabel → "Feedforward solution", AspectRatio → 1 / 3,
  ImageSize → 400, GridLines → {None, {-π, π}}]
p1b = Plot[{θ1b[t], u1a[t], x1b[t], θdot1b[t], xdot1b[t]},
  {t, 0, τ1}, PlotRange → {-4, 4}, Filling → {2 → Axis},
  PlotLegends → {"θ1b", "u1b", "x1b", "θdot1b", "xdot1b"}, PlotLabel → "Test on dynamics",
  AspectRatio → 1 / 3, ImageSize → 400, GridLines → {None, {-π, π}}]
p1c = Plot[{θ1c[t], u1c[t], x1c[t], θdot1c[t], xdot1c[t]}, {t, 0, τ1}, PlotRange → {-4, 4},
  Filling → {2 → Axis}, PlotLegends → {"θ1c", "u1c", "x1c", "θdot1c", "xdot1c"},
  PlotLabel → "LQR feedback solution", AspectRatio → 1 / 3,
  ImageSize → 400, GridLines → {None, {-π, π}}]
```

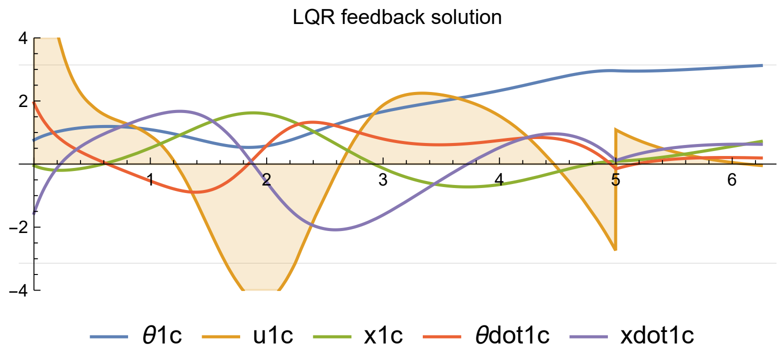
Out[1715]=



Out[1716]=



Out[1717]=



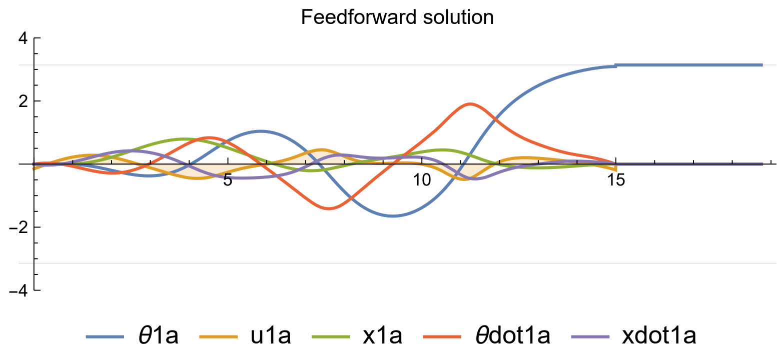
Increasing τ makes the problem much difficult to solve , we also need to increase the value of n

In[1548]:=
```
n = 40; τ = 15; τ1 = τ * 1.25; A = 0.2; order = 4; maxIter = 100; Weight = 0;
ICs = {-0.048555026816794494`,
   -1.560065966757075`, 0.7613152376955525`, 1.9382391342292873`};
ICs = {0, 0, 0, 0};
InitGuess =
  {RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}]};
InitGuess = {0, 0, 0, 0};
{x1a, xdot1a, θ1a, θdot1a, u1a, λ1ff0, λ2ff0, λ3ff0, λ4ff0} =
  ffCartPendulum[ICs, n, τ, τ1, A, order, maxIter, InitGuess, Weight];
{x1b, xdot1b, θ1b, θdot1b, u1b, J1} = testSwingUp[ICs, τ, τ1, u1a, A];
{x1c, xdot1c, θ1c, θdot1c, u1c, J} =
 testWithFB[ICs, τ, τ1, x1a, xdot1a, θ1a, θdot1a, u1a, A];
p1a = Plot[{θ1a[t], u1a[t], x1a[t], θdot1a[t], xdot1a[t]}, {t, 0, τ1}, Filling → {2 → Axis},
  PlotRange → {-4, 4}, PlotLegends → {"θ1a", "u1a", "x1a", "θdot1a", "xdot1a"},
  PlotLabel → "Feedforward solution", AspectRatio → 1 / 3,
  ImageSize → 400, GridLines → {None, {-π, π}}]
p1b = Plot[{θ1b[t], u1a[t], x1b[t], θdot1b[t], xdot1b[t]},
  {t, 0, τ1}, PlotRange → {-4, 4}, Filling → {2 → Axis},
  PlotLegends → {"θ1b", "u1b", "x1b", "θdot1b", "xdot1b"}, PlotLabel → "Test on dynamics",
  AspectRatio → 1 / 3, ImageSize → 400, GridLines → {None, {-π, π}}]
p1c = Plot[{θ1c[t], u1c[t], x1c[t], θdot1c[t], xdot1c[t]}, {t, 0, τ1}, PlotRange → {-4, 4},
  Filling → {2 → Axis}, PlotLegends → {"θ1c", "u1c", "x1c", "θdot1c", "xdot1c"},
  PlotLabel → "LQR feedback solution", AspectRatio → 1 / 3,
  ImageSize → 400, GridLines → {None, {-π, π}}]
```

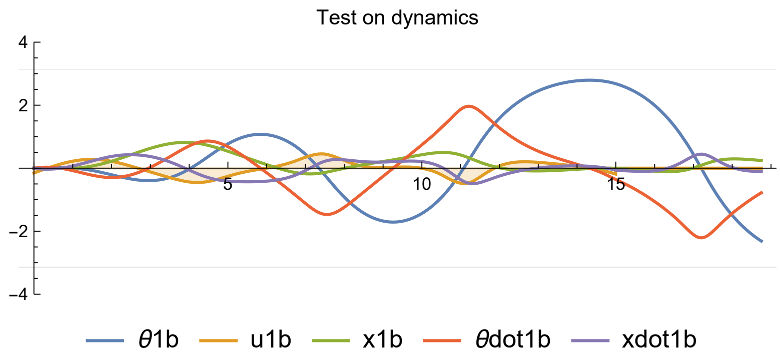
FindRoot: The line search decreased the step size to within tolerance specified by AccuracyGoal and PrecisionGoal but was unable to find a sufficient decrease in the merit function. You may need more than MachinePrecision digits of working precision to meet these tolerances.

NIntegrate: NIntegrate failed to converge to prescribed accuracy after 9 recursive bisections in t$485732 near {t$485732} = {11.6306}. NIntegrate obtained 0.8817100448459009` and 0.000011277560712045734` for the integral and error estimates.

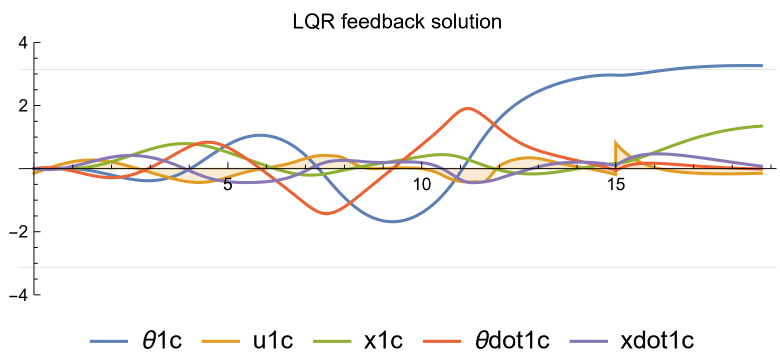
Out[1555]=



Out[1556]=



Out[1557]=

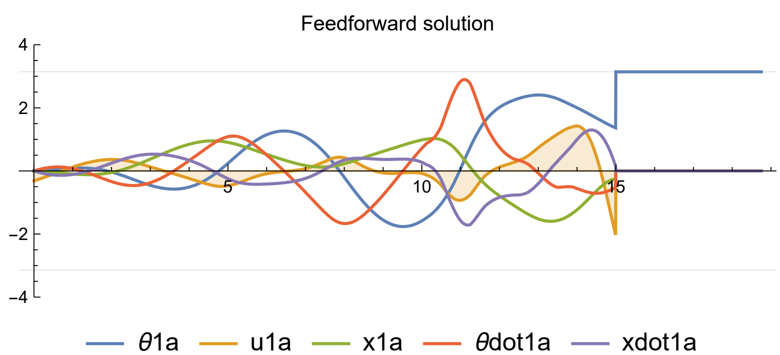


In[1718]:=
```
n = 40; τ = 15; τ1 = τ * 1.25; A = 0.2; order = 4; maxIter = 100; Weight = 0.01;
ICs = {-0.048555026816794494`,
   -1.560065966757075`, 0.7613152376955525`, 1.9382391342292873`};
ICs = {0, 0, 0, 0};
InitGuess =
  {RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}]};
InitGuess = {0, 0, 0, 0};
{x1a, xdot1a, θ1a, θdot1a, u1a, λ1ff0, λ2ff0, λ3ff0, λ4ff0} =
  ffCartPendulum[ICs, n, τ, τ1, A, order, maxIter, InitGuess, Weight];
{x1b, xdot1b, θ1b, θdot1b, u1b, J1} = testSwingUp[ICs, τ, τ1, u1a, A];
{x1c, xdot1c, θ1c, θdot1c, u1c, J} =
 testWithFB[ICs, τ, τ1, x1a, xdot1a, θ1a, θdot1a, u1a, A];
p1a = Plot[{θ1a[t], u1a[t], x1a[t], θdot1a[t], xdot1a[t]}, {t, 0, τ1}, Filling → {2 → Axis},
  PlotRange → {-4, 4}, PlotLegends → {"θ1a", "u1a", "x1a", "θdot1a", "xdot1a"},
  PlotLabel → "Feedforward solution", AspectRatio → 1 / 3,
  ImageSize → 400, GridLines → {None, {-π, π}}]
p1b = Plot[{θ1b[t], u1a[t], x1b[t], θdot1b[t], xdot1b[t]},
  {t, 0, τ1}, PlotRange → {-4, 4}, Filling → {2 → Axis},
  PlotLegends → {"θ1b", "u1b", "x1b", "θdot1b", "xdot1b"}, PlotLabel → "Test on dynamics",
  AspectRatio → 1 / 3, ImageSize → 400, GridLines → {None, {-π, π}}]
p1c = Plot[{θ1c[t], u1c[t], x1c[t], θdot1c[t], xdot1c[t]}, {t, 0, τ1}, PlotRange → {-4, 4},
  Filling → {2 → Axis}, PlotLegends → {"θ1c", "u1c", "x1c", "θdot1c", "xdot1c"},
  PlotLabel → "LQR feedback solution", AspectRatio → 1 / 3,
  ImageSize → 400, GridLines → {None, {-π, π}}]
```

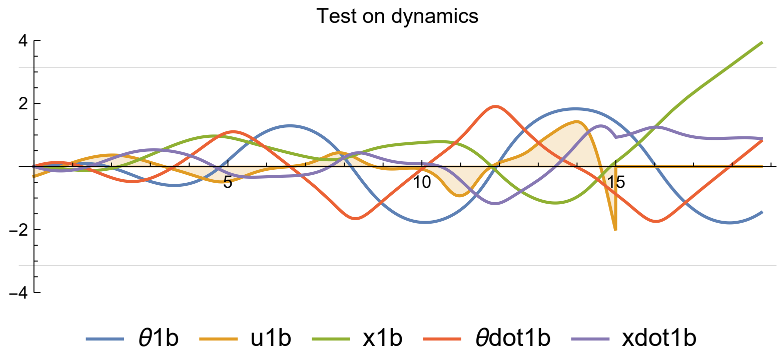
FindRoot: Failed to converge to the requested accuracy or precision within 100 iterations.

NIntegrate: NIntegrate failed to converge to prescribed accuracy after 9 recursive bisections in t$604163 near {t$604163} = {12.7732}. NIntegrate obtained 1152.97213499595` and 0.0049085104068194935` for the integral and error estimates.

Out[1725]=



Out[1726]=



Out[1727]=