In[2199]:=

```
ClearAll["Global`*"];
Remove["Global`*"];
(*ICs - Initial Conditions *) (* Error while cosntraining u *)
ffCartPendulum[ICs_,n_,τ_,τ1_,A_,order_,maxIter_,maxError_]:=
Module[{InitGuess,error,x,dist,xdot,f,θ,θdot,λ1,λ2,λ3,λ4,Δt,bcs,eqns,sv,froot,xff,xdotff,xff0,xdo
f[{x_,xdot_,θ_,θdot_,λ1_,λ2_,λ3_,λ4_}]:={
    xdot,
    1/(1-A Cos[θ]^2) (A θdot^2 Sin[θ]+1/(1-A Cos[θ]^2) (λ4 Cos[θ]-λ2)+A Cos[θ] Sin[θ]),
    θdot,
    1/(1-A Cos[θ]^2) (-(1/(1-A Cos[θ]^2))(-λ2 Cos[θ]+λ4 Cos[θ]^2)-Sin[θ]-A θdot^2 Cos[θ] Sin[θ
    0,
    -λ1,
    1/(-1+A Cos[θ]^2)^3 (A^2 (A θdot^2 λ2-λ4) Cos[θ]^5+A^3 (λ2-θdot^2 λ4) Cos[θ]^6-1/2 A^2 (λ2-θdot^2 λ4) Cos[
    4 /(A Cos[2 θ]+A-2) (A θdot Sin[θ] (λ2-λ4 Cos[θ]))-λ3
};

InitGuess = {0,0,0,0};
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,Subs
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];

error = Norm[Flatten[Join[{x_n,xdot_n,θ_n,θdot_n} - {0,0,π,0},{x_0,xdot_0,θ_0,θdot_0} - ICs,Table[Thread[

While[error > maxError,
InitGuess = {RandomReal[{-1,1}],RandomReal[{-1,1}],RandomReal[{-1,1}],RandomReal[{-1,1}]};
xGuess = Table[If[i ≠ -1,xGuess_{i+1} = xGuess_i +Δt*f[xGuess_i],xGuess_0 = {ICs⟦1⟧,ICs⟦2⟧,ICs⟦3⟧,ICs⟦4
sv = Flatten[Table[{{Subscript[x, i],xGuess⟦i+1⟧⟦1⟧},{Subscript[xdot, i],xGuess⟦i+1⟧⟦2⟧},{Subsc
                    {Subscript[λ1, i],xGuess⟦i+1⟧⟦5⟧},{Subscript[λ2, i],xGuess⟦i+1⟧⟦6⟧},{Subscr

froot=FindRoot[eqns,sv,MaxIterations→maxIter];

error = Norm[Flatten[Join[{x_n,xdot_n,θ_n,θdot_n} - {0,0,π,0},{x_0,xdot_0,θ_0,θdot_0} - ICs,Table[Thread[

];



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;
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]));
```

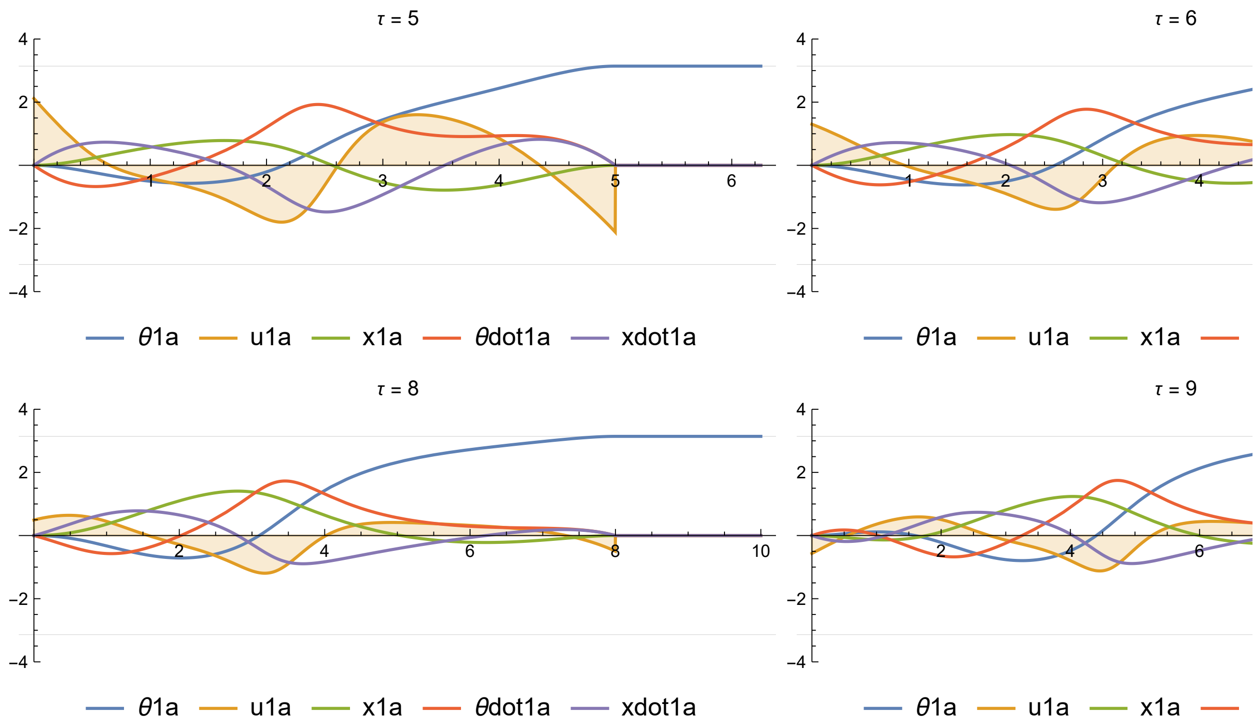
```
fx = {xdot,x2dot,θdot,θ2dot};
Bf = D[fx,u] ;(*For 1D stuff use D*)
K = (Bfᵀ.S[τ - time])/. {x→ x1a[time], xdot → xdot1a[time], θ →θ1a[time], θdot → θdot1a[time],
K
]
testWithFB[ICs_,τ_,τ1_,xff0_,xdotff0_,θff0_,θdotff0_,uff0_,A_]:=Module[{eq,init,θ,θdot,θff,θdot
κ1=κ2=3;  (* lqr for q=r for balancing 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}]
```

Now we choose a large enough n and vary the time horizon to understand the behaviour of u. Fix an initial condition

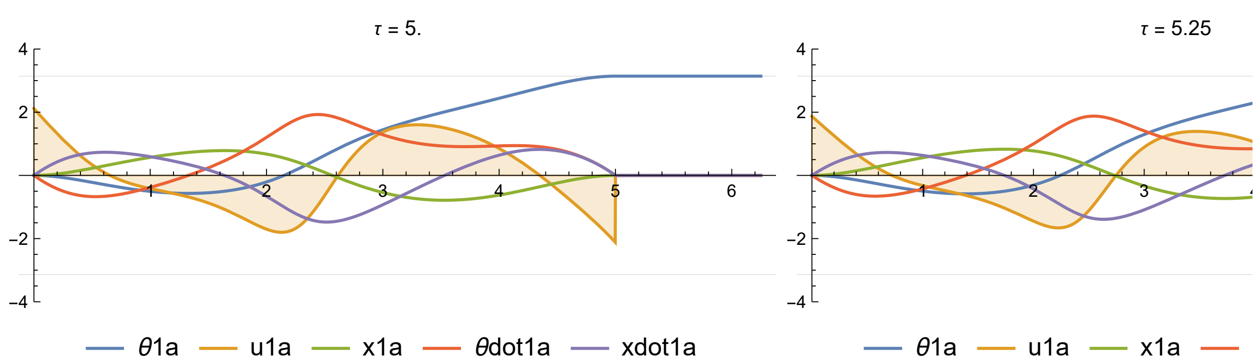
In[2294]:=
```
PlotFF[n_, τ_, A_, order_, maxIter_, maxError_, ICs_] :=
 Module[{τ1, plot, x1a, xdot1a, θ1a, θdot1a, u1a, λ1ff0, λ2ff0, λ3ff0, λ4ff0},
  τ1 = 1.25 * τ;
  {x1a, xdot1a, θ1a, θdot1a, u1a, λ1ff0, λ2ff0, λ3ff0, λ4ff0} =
   Quiet[ffCartPendulum[ICs, n, τ, τ1, A, order, maxIter, maxError]]; plot =
   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 → StringJoin["τ = ", ToString[τ]] , AspectRatio → 1 / 3,
    ImageSize → 400, GridLines → {None, {-π, π}}];
  plot]
```

In[2349]:=
```
n = 60; τ = 5; τ1 = τ * 1.25; A = 0.2; order = 4; maxIter = 30; maxError = 0.01;
ICs = {0, 0, 0, 0};
τStart = 5; τEnd = 10; τStep = 1;
numberPlots = IntegerPart[(τEnd - τStart) / (τStep)] + 1;
plots = Table[PlotFF[n, τ, A, order, maxIter, maxError, ICs], {τ, τStart, τEnd, τStep}];
Grid[Join[Table[Table[plots[[i]], {i, j, j + 2}], {j, 1, 3 * IntegerPart[numberPlots / 3], 3}],
  Table[Table[plots[[i]], {i, 3 * IntegerPart[numberPlots / 3] + 1, numberPlots}], {j, 1, 1}]]]
```

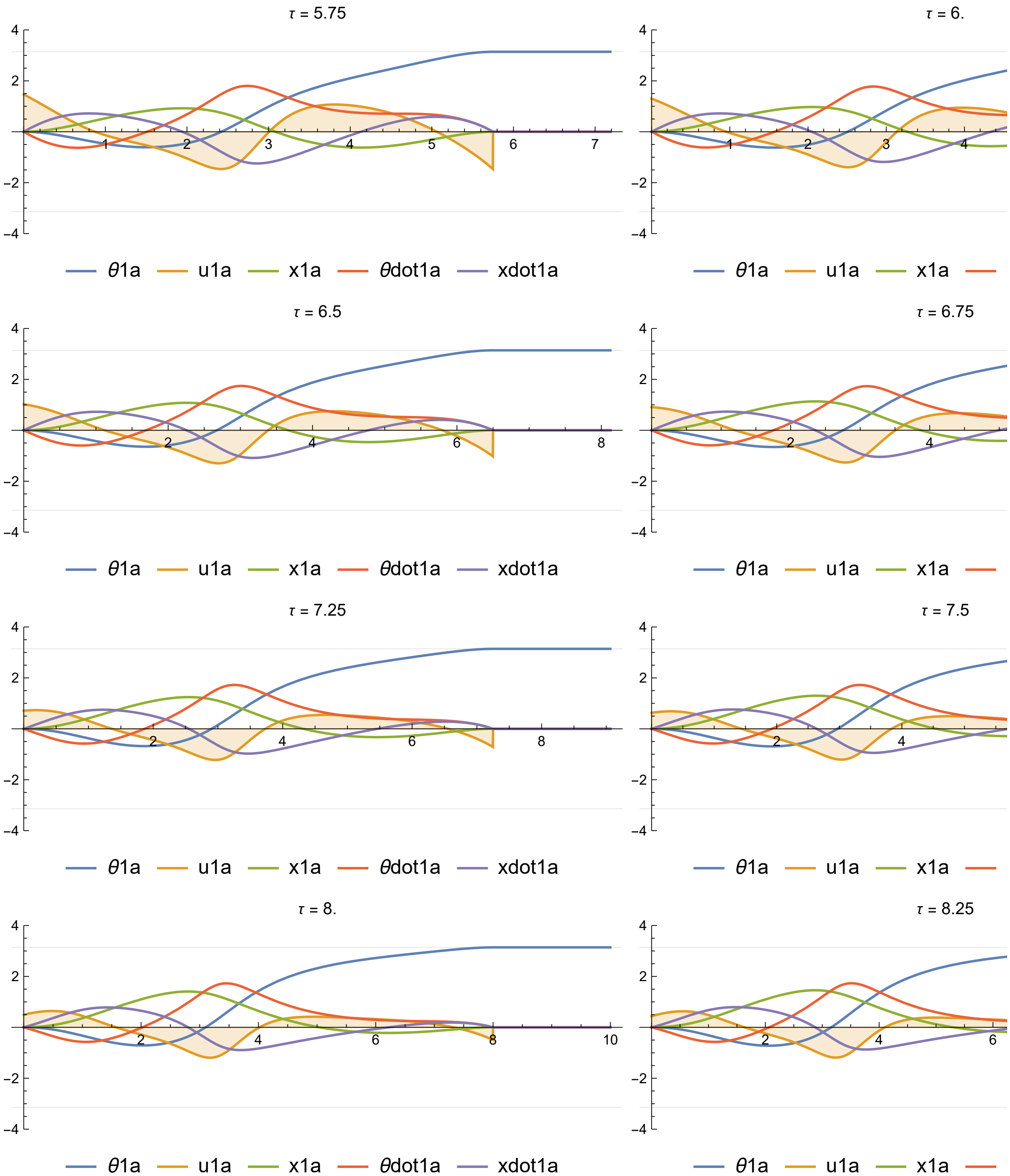
Out[2353]=

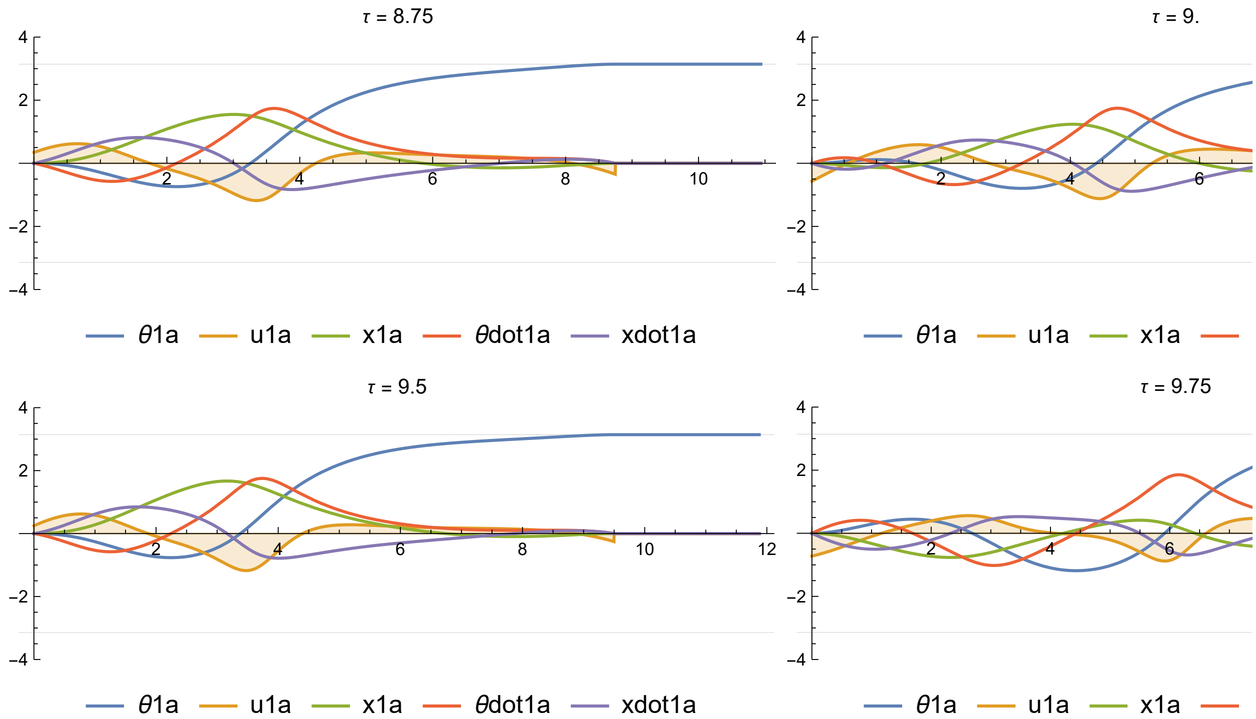


In[2354]:=
```
n = 60; τ = 5; τ1 = τ * 1.25; A = 0.2; order = 4; maxIter = 30; maxError = 0.01;
ICs = {0, 0, 0, 0};
τStart = 5; τEnd = 10; τStep = 0.25;
numberPlots = IntegerPart[(τEnd - τStart) / (τStep)] + 1;
plots = Table[PlotFF[n, τ, A, order, maxIter, maxError, ICs], {τ, τStart, τEnd, τStep}];
Grid[Join[Table[Table[plots[[i]], {i, j, j + 2}], {j, 1, 3 * IntegerPart[numberPlots / 3], 3}],
  Table[Table[plots[[i]], {i, 3 * IntegerPart[numberPlots / 3] + 1, numberPlots}], {j, 1, 1}]]]
```



Out[2358]=


τ = 5.75
θ1a u1a x1a θdot1a xdot1a
τ = 6.
θ1a u1a x1a
τ = 6.5
θ1a u1a x1a θdot1a xdot1a
τ = 6.75
θ1a u1a x1a
τ = 7.25
θ1a u1a x1a θdot1a xdot1a
τ = 7.5
θ1a u1a x1a
τ = 8.
θ1a u1a x1a θdot1a xdot1a
τ = 8.25
θ1a u1a x1a


τ = 8.75
θ1a
u1a
x1a
θdot1a
xdot1a
τ = 9.
θ1a
u1a
x1a
τ = 9.5
θ1a
u1a
x1a
θdot1a
xdot1a
τ = 9.75
θ1a
u1a
x1a