**Code:**

function [PInf SigInf gam] = HW13RicCalc(gamStart,num)

%Define variables from prob, calc oth. mats. needed

%Note Z = Y^-1 = eye, X = eye

Currentgam = gamStart;

Oldgam = gamStart\*2;

for k = 1:num

gam = Currentgam;

fprintf('The current gamma being used is %d.\n',gam)

%Find P\*

options = bvpset('RelTol', 1e-6);

solinit = bvpinit(linspace(0,20),[60;25;25;10]);

PSol = bvp4c(@(t,P)POdeFunc(t,P,A,B,M,Q,R,S,gam,-1),...

@PBCFunc,solinit,options);

PInf = PSol.y(:,1);

PInf = [PInf(1:2),PInf(3:4)];

%Find Sig\*

SigSol = ode45(@(t,Sig)SigOdeFunc(t,Sig,A,C,L,M,N,Q,gam),...

[0,10],[1;0;0;1]);

SigInf = SigSol.y(:,end);

SigInf = [SigInf(1:2),SigInf(3:4)];

%Check conditions

kNow = abs(eigs(SigInf\*PInf,1));

if kNow < gam^2

Oldgam = Currentgam;

Currentgam = Currentgam/2;

fprintf('The current gam value is greater then gam opt.\n')

else

Currentgam = (Oldgam+Currentgam)/2;

fprintf('The current gam value is less then gam opt.\n')

end

end

%Functions for calculating P\* (sol to ARE). BVCProb.

function dP = POdeFunc(~,P,A,B,M,Q,R,S,gam,sign)

P = P(:);

P = [P(1:2),P(3:4)];

dP = -sign\*(-P\*A - A'\*P + (P\*B+S)\*(R^-1)\*(B'\*P+S')-...

P\*M\*P/gam^2 - Q);

dP = [dP(:,1);dP(:,2)];

function res = PBCFunc(~,yb)

res = [yb(1)-1;...

yb(2)-0;...

yb(3)-0;...

yb(4)-1];

%Function for calculating Sig\*

function dSig = SigOdeFunc(~,Sig,A,C,L,M,N,Q,gam)

Sig = Sig(:);

Sig = [Sig(1:2),Sig(3:4)];

dSig = A\*Sig + Sig\*A' - (Sig\*C'+L')\*(N^-1)\*(C\*Sig+L)+...

Sig\*Q\*Sig/gam^2 + M;

dSig = [dSig(:,1);dSig(:,2)];