Upper bound demonstrations

Contents

VDP with deg(V) = 2

Form the vector field

```
pvar x1 x2;
x = [x1;x2];
x1dot = -x2;
x2dot = x1+(x1^2-1)*x2;
f = [x1dot; x2dot];
```

Get the default values of options to run the ROA code.

```
zV = monomials(x,2:2);
Bis.r1deg = 2;
```

Now, call GetRoaOpts to get the correponding opts, roaconstr, etc.

```
[roaconstr,opt,sys] = GetRoaOpts(f, x, zV, [],Bis);
opt.sim.NumConvTraj = 200;
opt.sim.dispEveryNth = 40;
opt.display.roaest = 1;
opt.coordoptim.IterStopTol = 1e-4;
```

Call the wrapper which in turn calls RoaEst.m

```
outputs = wrapper(sys,[],roaconstr,opt);
```

```
------------------Beginning simulations
System 1: Num Stable = 0        Num Unstable = 1        Beta for Sims = 2.348   Beta UB = 2.348
System 1: Num Stable = 40       Num Unstable = 1        Beta for Sims = 2.348   Beta UB = 2.348
System 1: Num Stable = 43       Num Unstable = 2        Beta for Sims = 2.230   Beta UB = 2.348
System 1: Num Stable = 80       Num Unstable = 2        Beta for Sims = 2.230   Beta UB = 2.348
System 1: Num Stable = 120      Num Unstable = 2        Beta for Sims = 2.230   Beta UB = 2.348
System 1: Num Stable = 160      Num Unstable = 2        Beta for Sims = 2.230   Beta UB = 2.348
System 1: Num Stable = 200      Num Unstable = 2        Beta for Sims = 2.230   Beta UB = 2.348
------------------End of simulations
------------------Begin search for feasible V
Try = 1          Beta for Vfeas = 2.230
Try = 2          Beta for Vfeas = 2.119
------------------Found feasible V
Initial V (from the cvx outer bnd) gives Beta = 1.496
-------------------Iteration = 1
Beta = 1.513 (Gamma = 0.746)
-------------------Iteration = 2
Beta = 1.516 (Gamma = 0.747)
-------------------Iteration = 3
Beta = 1.517 (Gamma = 0.747)
-------------------Iteration = 4
Beta = 1.517 (Gamma = 0.747)
```

Extract the solution

```
[V,beta,gamma,p,multip,betaUpper] = extractSol(outputs);
```

Upper bounds from divergent trajectories

```
betaUpperDivergent = outputs.RoaEstInfo.info.SimLFG.sim.BetaUB;
betaUpperDivergent
```

```
betaUpperDivergent =

    2.3478
```

Upper bound from infeasibility of the relaxation

```
if betaUpper < outputs.RoaEstInfo.info.SimLFG.sim.BetaUB
    betaUpperInfeas = betaUpper;
    betaUpperInfeas
else
    display('No upper bound from infeasibility');
end
```

```
betaUpperInfeas =

    2.2304
```

Certified beta

```
betaCertified = beta;
betaCertified
```

```
betaCertified =

    1.5168
```

VDP with deg(V) = 6

Form the vector field

```
pvar x1 x2;
x = [x1;x2];
x1dot = -x2;
x2dot = x1+(x1^2-1)*x2;
f = [x1dot; x2dot];
```

Get the default values of options to run the ROA code.

```
zV = monomials(x,2:6);
Bis.r1deg = 4;
```

Now, call GetRoaOpts to get the correponding opts, roaconstr, etc.

```
[roaconstr,opt,sys] = GetRoaOpts(f, x, zV, [],Bis);
opt.sim.NumConvTraj = 200;
opt.sim.dispEveryNth = 40;
opt.display.roaest = 1;
opt.coordoptim.IterStopTol = 1e-4;
```

Call the wrapper which in turn calls RoaEst.m

```
outputs = wrapper(sys,[],roaconstr,opt);
```

```
------------------Beginning simulations
System 1: Num Stable = 0        Num Unstable = 1        Beta for Sims = 2.352   Beta UB = 2.352
System 1: Num Stable = 7        Num Unstable = 2        Beta for Sims = 2.235   Beta UB = 2.347
System 1: Num Stable = 40       Num Unstable = 2        Beta for Sims = 2.235   Beta UB = 2.347
System 1: Num Stable = 80       Num Unstable = 2        Beta for Sims = 2.235   Beta UB = 2.347
System 1: Num Stable = 120      Num Unstable = 2        Beta for Sims = 2.235   Beta UB = 2.347
System 1: Num Stable = 160      Num Unstable = 2        Beta for Sims = 2.235   Beta UB = 2.347
System 1: Num Stable = 200      Num Unstable = 2        Beta for Sims = 2.235   Beta UB = 2.347
------------------End of simulations
------------------Begin search for feasible V
Try = 1          Beta for Vfeas = 2.235
------------------Found feasible V
Initial V (from the cvx outer bnd) gives Beta = 0.583
-------------------Iteration = 1
Beta = 1.436 (Gamma = 1.387)
-------------------Iteration = 2
Beta = 1.731 (Gamma = 1.618)
-------------------Iteration = 3
Beta = 1.886 (Gamma = 1.770)
-------------------Iteration = 4
Beta = 1.981 (Gamma = 1.874)
-------------------Iteration = 5
Beta = 2.049 (Gamma = 1.948)
-------------------Iteration = 6
Beta = 2.098 (Gamma = 2.001)
-------------------Iteration = 7
Beta = 2.134 (Gamma = 2.041)
-------------------Iteration = 8
Beta = 2.164 (Gamma = 2.072)
-------------------Iteration = 9
Beta = 2.188 (Gamma = 2.098)
-------------------Iteration = 10
Beta = 2.209 (Gamma = 2.119)
-------------------Iteration = 11
Beta = 2.228 (Gamma = 2.137)
-------------------Iteration = 12
Beta = 2.244 (Gamma = 2.153)
-------------------Iteration = 13
Beta = 2.259 (Gamma = 2.166)
-------------------Iteration = 14
Beta = 2.272 (Gamma = 2.178)
-------------------Iteration = 15
Beta = 2.283 (Gamma = 2.188)
-------------------Iteration = 16
Beta = 2.293 (Gamma = 2.197)
-------------------Iteration = 17
Beta = 2.301 (Gamma = 2.204)
-------------------Iteration = 18
Beta = 2.308 (Gamma = 2.209)
-------------------Iteration = 19
Beta = 2.313 (Gamma = 2.213)
-------------------Iteration = 20
Beta = 2.317 (Gamma = 2.217)
```

Extract the solution

```
[V,beta,gamma,p,multip,betaUpper] = extractSol(outputs);
```

Upper bounds from divergent trajectories

```
betaUpperDivergent = outputs.RoaEstInfo.info.SimLFG.sim.BetaUB;
betaUpperDivergent
```

```
betaUpperDivergent =

    2.3470
```

Upper bound from infeasibility of the relaxation

```
if betaUpper < outputs.RoaEstInfo.info.SimLFG.sim.BetaUB
    betaUpperInfeas = betaUpper;
    betaUpperInfeas
else
    display('No upper bound from infeasibility');
end
```

```
No upper bound from infeasibility
```

Certified beta

```
betaCertified = beta;
betaCertified
```

```
betaCertified =

    2.3173
```

Published with MATLAB® 7.6