Robust ROA calculations

dynamics:

x1dot = x2;

x2dot = -x2-2*x1+2*x1^3 + delta*(-x1+x1^3);

with delta \in [-1,1]

This example was also used in Topcu and Packard, IEEE TAC, 2009 (in the special issue on positive polynomials in controls (example 1 in the paper)

```
% Form the vector field
pvar x1 x2;
x = [x1;x2];
x1dot = x2;
x2dot = -x2-2*x1+2*x1^3;
```

Nominal system

```
f = [x1dot; x2dot];
```

Introduce an uncertain parameter

```
pvar d1
```

Specify its range

```
ini_cell = [-1 1];
```

Form the uncertain vector field

```
f = f + d1*[0; -x1+x1^3];
```

```
% Get the vertex system
[roaconstr,opt,sys] = GetRoaOpts(f, x);
[fNOM,fVER] = getf(sys,ini_cell);

% Generate the options, etc.
zV = monomials(x,2:4);
Bis.flag = 0;
Bis.r1deg = 4;
```

```
[roaconstr,opt,sys] = GetRoaOpts(fVER, x, zV, [], Bis);
sys.fWithDel = [];

opt.sim.NumConvTraj = 40;
opt.display.roaest = 1;
```

Run the computations

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

```
------------------Beginning simulations
System 1: Num Stable = 0        Num Unstable = 1        Beta for Sims = 3.289   Beta UB = 3.289
System 1: Num Stable = 0        Num Unstable = 2        Beta for Sims = 1.390   Beta UB = 1.390
System 1: Num Stable = 2        Num Unstable = 3        Beta for Sims = 1.306   Beta UB = 1.306
System 1: Num Stable = 4        Num Unstable = 4        Beta for Sims = 0.913   Beta UB = 0.913
System 1: Num Stable = 6        Num Unstable = 5        Beta for Sims = 0.861   Beta UB = 0.861
System 1: Num Stable = 12       Num Unstable = 6        Beta for Sims = 0.818   Beta UB = 0.842
System 1: Num Stable = 18       Num Unstable = 7        Beta for Sims = 0.777   Beta UB = 0.808
System 2: Num Stable = 1        Num Unstable = 1        Beta for Sims = 1.476   Beta UB = 0.808
System 2: Num Stable = 3        Num Unstable = 2        Beta for Sims = 1.402   Beta UB = 0.808
System 2: Num Stable = 6        Num Unstable = 3        Beta for Sims = 1.114   Beta UB = 0.808
System 2: Num Stable = 6        Num Unstable = 4        Beta for Sims = 1.058   Beta UB = 0.808
System 2: Num Stable = 8        Num Unstable = 5        Beta for Sims = 1.000   Beta UB = 0.808
System 2: Num Stable = 10       Num Unstable = 6        Beta for Sims = 0.929   Beta UB = 0.808
System 2: Num Stable = 11       Num Unstable = 7        Beta for Sims = 0.882   Beta UB = 0.808
------------------End of simulations
------------------Begin search for feasible V
Try = 1          Beta for Vfeas = 0.882
Try = 2          Beta for Vfeas = 0.838
------------------Found feasible V
Initial V (from the cvx outer bnd) gives Beta = 0.173
-------------------Iteration = 1
Beta = 0.567 (Gamma = 0.535)
-------------------Iteration = 2
Beta = 0.665 (Gamma = 0.604)
-------------------Iteration = 3
Beta = 0.716 (Gamma = 0.640)
-------------------Iteration = 4
Beta = 0.739 (Gamma = 0.656)
```

Extract the solution

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

```
beta
```

```
beta =

    0.7388
```

Upper bound on beta

```
betaUpper
```

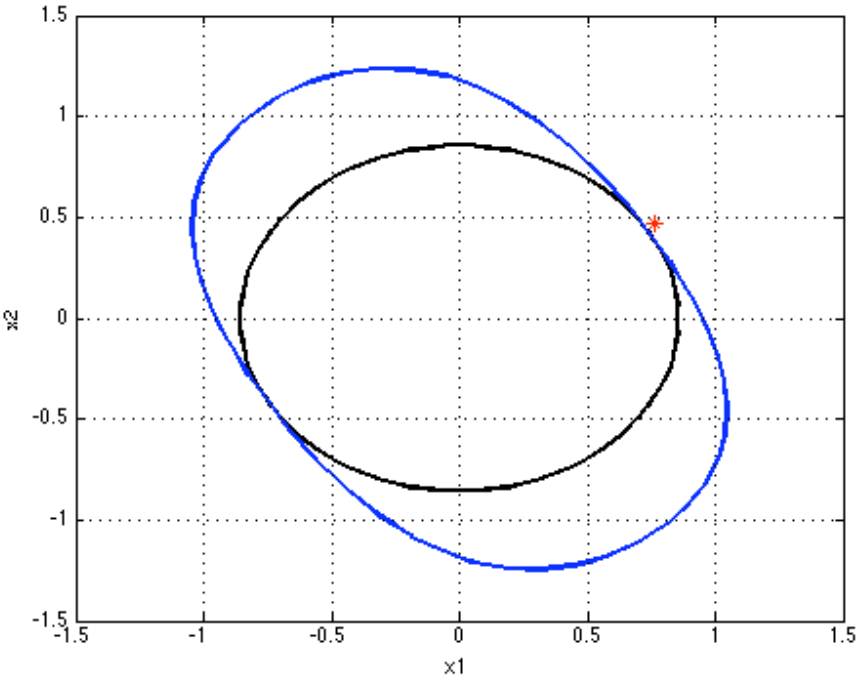
```
betaUpper =

    0.8822
```

Plot the results

```
[Cp4,hp4] = pcontour(p,beta,[-2 2 -2 2],'k'); hold on;
set(hp4,'linewidth',2);
[CV4,hV4] = pcontour(V,gamma,[-2 2 -2 2],'b');
set(hV4,'linewidth',2);
set(gca,'xlim',[-1.5 1.5],'ylim',[-1.5 1.5]);

traj = outputs.RoaEstInfo.info.SimLFG.sim.Trajectories(1).unstab(end).state;
pval = peval(traj,p.coef,p.deg);
[aux,ind] = min(pval);
plot(traj(1,ind),traj(2,ind),'r*','markersize',8);
grid on;
```



Published with MATLAB® 7.6