ROA computation for system with unmodeled dynamics

This code demonstrates the robust ROA calculations for systems with unmodeled dynamics on the controlled aircraft dynamics where the unmodeled dynamics connect to the nominal model as shown in slide titled "Example: Controlled aircraft dynamics with unmodeled dynamics" in section "Robust ROA and performance analysis with unmodeled dynamics"

Form the input out dynamics (w-->z) with

xdot = f(x,w)

z = h(x)

```
pvar x1 x2 x3 x4 w

x = [x1;x2;x3;x4];
y = [x1;x3];
Cc = 2;
v = Cc*x4;
Ac = [-0.864 -0.3211];
x4dot = Ac*y;
u = 1.25*v + w;
f(1,1) = -0.24366*x2^3 + 0.082272*x1*x2 + 0.30492*x2^2 - 0.082272*x2*u/2 +
f(2,1) = -0.054444*x2^2 + 0.10889*x2*x3 - 0.054444*x3^2 + 0.91136*x1 - 0.6
f(3,1) = x1;
f(4,1) = x4dot;
h = 0.75*Cc*x4;
```

The unmodeled dynamics has gain <= 1

```
gainUnmodeled = 1;
```

Generate some options used in the iterations

V - l1 is SOS

-((Beta-p)*sB - (R2-V)) is SOS

-((R2-V)*sV + Vdot - w'*w + z'*z/gamma^2) is SOS

with sB and sV SOS

```
rVgCons.p = x'*x;
rVgCons.l1 = 1e-6*x'*x;
```

basis for multipliers

```
rVgCons.sVBasis = monomials([x;w],1:1);
rVgCons.sBBasis = monomials([x],0:0);
```

basis for V

```
rVgCons.VBasis = monomials(x,2:2);
```

Options used in the optimization (there are two types of bisection)

Options for R2 bisection (given V maximize R2)

```
opts.OptBis.R2High = 100;
opts.OptBis.R2Low = 0;
opts.OptBis.R2Tol = 1e-5;
```

Options for beta bisection (given V and R2 maximize beta)

```
opts.OptBis.BetaLow = 0;
opts.OptBis.BetaHigh = 10;
opts.OptBis.BetaTol = 1e-5;
```

Options for outer iterations

```
opts.MaxIter = 30;
opts.StopTol = 1e-4;
```

Call the routine for the robust ROA calculation.

```
[bOut,ROut,VOut,sVOut,sBOut] =...
              roaViaGain(f,h,x,w,gainUnmodeled, rVgCons, opts);
```

```
--Iteration = 1; Beta = 2.237864e+00
--Iteration = 2; Beta = 2.528324e+00
--Iteration = 3; Beta = 2.674789e+00
--Iteration = 4; Beta = 2.766418e+00
--Iteration = 5; Beta = 2.841692e+00
--Iteration = 6; Beta = 2.909317e+00
--Iteration = 7; Beta = 2.969599e+00
--Iteration = 8; Beta = 3.024187e+00
--Iteration = 9; Beta = 3.073797e+00
--Iteration = 10; Beta = 3.119020e+00
--Iteration = 11; Beta = 3.160076e+00
```

```
--Iteration = 12; Beta = 3.198042e+00
--Iteration = 13; Beta = 3.232651e+00
--Iteration = 14; Beta = 3.264246e+00
--Iteration = 15; Beta = 3.293161e+00
--Iteration = 16; Beta = 3.319778e+00
--Iteration = 17; Beta = 3.344307e+00
--Iteration = 18; Beta = 3.366671e+00
--Iteration = 19; Beta = 3.387337e+00
--Iteration = 20; Beta = 3.406572e+00
--Iteration = 21; Beta = 3.424530e+00
--Iteration = 22; Beta = 3.441334e+00
--Iteration = 23; Beta = 3.457098e+00
--Iteration = 24; Beta = 3.471813e+00
--Iteration = 25; Beta = 3.485594e+00
--Iteration = 26; Beta = 3.498507e+00
--Iteration = 27; Beta = 3.510599e+00
--Iteration = 28; Beta = 3.521967e+00
--Iteration = 29; Beta = 3.532410e+00
--Iteration = 30; Beta = 3.542099e+00
```

Conclusion: For any system Phi which is known to be strictly dissipative w.r.t. z^Tz-w^Tw, if the Phi dunamics have zero initial conditions, then, for any x(0) in { x : p(x(0)) \leq bOut}, x(t) stays in { x : V(x) \leq ROut^2} and x(t) --> 0 as t--> infty where

```
bOut
```

```
bOut =

    3.5421
```

```
ROut
```

```
ROut =

    5.3042
```

```
VOut
```

```
VOut =
  0.42274*x1^2
  - 0.30165*x1
  *x2 + 0.72089
  *x1*x3
  - 0.89644*x1
  *x4 + 3.7846
  *x2^2
  - 6.3846*x2
  *x3 - 0.13911
  *x2*x4
  + 5.1233
  *x3^2
  + 2.3659*x3
  *x4 + 3.6094
  *x4^2
```