Sum-of-Squares Programming

This script introduces polynomial optimization using sum-of-squares (SOS) programming.

Contents

Requirements

This script requires the SOSAnalysis toolbox: http://www.aem.umn.edu/~AerospaceControl/

Optimizing over a single SOS constraint

Find the minimum value of a such that

$$p = x^2 - axy^2 + 2x^4 + bx^3y - 2x^2y^2 + 6y^4$$

is SOS in the variables (x, y).

Create Polynomial and Variables

```
pvar a b x y
p = x^2 - a*x*y^2 + 2*x^4 + b*x^3*y - 2*x^2*y^2 + 6*y^4;

% Independent variables
ind_vars = [x; y];
```

Minimization using sosopt

Any variables (in p) that are not listed in the 2nd argument of sosopt are treated as decision variables for the minimization. Therefore, in this example both a and b are decision variables.

```
[info,dopt,sossol] = sosopt(p, ind_vars, a);
dopt

% Substitute the optimal values for a and b into p
p_ab = subs(p, dopt)

% Check Results
```

```
issos(p_ab)
```

```
dopt =
  [ a,     -4.690415755919728]
  [ b, -1.093037714707311e-15]


p_ab =
  2*x^4 - 1.093037714707311e-15*x^3*y - 2*x^2*y^2 + 6*y^4
  + 4.690415755919728*x*y^2 + x^2


ans =

     1
```

Maximization using sosopt

Maximize a by minimizing $-a$

```
[info,dopt,sossol] = sosopt(p, ind_vars, -a);
dopt

% Substitute the optimal values for a and b into p
p_ab = subs(p, dopt)

% Check Results
issos(p_ab)
```

```
dopt =
  [ a,      4.690415755919728]
  [ b, -1.093037714707311e-15]


p_ab =
  2*x^4 - 1.093037714707311e-15*x^3*y - 2*x^2*y^2 + 6*y^4
  - 4.690415755919728*x*y^2 + x^2


ans =

     1
```

Optimizing over Multiple SOS Constraints

Find the minimum value of a such that

$p = 2x^2 + axy + 10y^2 + 2xy + y^2$

$$p_1 = 3x^2 + axy + 10y^2 + 3yz + z^2$$

and

$$p_2 = 6x^2 - 7xy - 3xz + (5-a)y^2 + 3yz + z^2$$

are SOS in the variables (x, y, z).

```
% Create polynomials
pvar a x y z
p1 = 3*x^2 + a*x*y + 10*y^2 + 3*y*z + z^2;
p2 = 6*x^2 - 7*x*y - 3*x*z + (5-a)*y^2 + 3*y*z + z^2;

% Independent variables
ind_vars = [x; y; z];

% SOS Constraints
sos_constraint = [p1; p2];
[info,dopt] = sosopt(sos_constraint, ind_vars, -a);
dopt
```

```
dopt =
  [ a, 2.333333345112194]
```

Check Results

```
[issos(subs(p1, dopt)) issos(subs(p2, dopt))]
```

```
ans =

     1     1
```

Conclusion

This example provides a brief overview of SOS optimization using the SOSAnalysis toolbox.

Attribution

This example supplements the book "Networks of Dissipative Systems: Compositional Certification of Stability, Performance, and Safety" by Murat Arcak, Chris Meissen, and Andrew Packard.

Published with MATLAB® R2015a