L2 Gain Certification of a Nonlinear System

This example determines a bound on the L2 gain of a nonlinear, polynomial system given by

$\dot{x} = -x + 1.5x^2 - x^3 + u$

$y = x + 0.3x^2$

By way of comparison, the Jacobian linearization is

$\dot{y}(t) + y(t) = u(t)$

which has an L2 gain of 1, serving as a lower bound to the L2 gain of the nonlinear system.

Contents

Requirements

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

Define Nonlinear System

Create `polynomial` variables

```
pvar x u;
```

Define system dynamics

```
f = -x + 1.5*x^2 - x^3 + u;
h = x + 0.3*x^2;
```

Create storage function

```
% Create 4th order storage function V (using quadratic, cubic and quartic
% terms) with coefficients "a" serving as decision variables.
Vmonomials = monomials(x, 2:4);
V = polydecvar('a', Vmonomials)
```

```
V =
  a_3*x^4 + a_2*x^3 + a_1*x^2
```

Specify SOS constraints

Create an array of expressions that are constrained to be sum-of-squares in the independent variables `(x,u)`. The first such constraint enforces the nonnegativity of V.

```
sos_constraint(1) = V;
```

The dissipation inequality

$$-\frac{dV}{dx} f(x,u) - h(x)^2 + \gamma^2 u^2 \geq 0 \quad \forall x \in R^n, u \in R^m$$

certifies the L2 gain is less than or equal to γ. Define a new variable `alpha` for `gamma^2`, and enforce the dissipation inequality as an SOS constraint.

```
pvar alpha
sos_constraint(2) = -jacobian(V,x)*f - h^2 + alpha*u^2;
```

Solve SOS problem

```
% Minimize alpha subject to the SOS constraints in the variables |(x,u)|
objective = alpha;
indVar = [x;u];
[info,dopt] = sosopt(sos_constraint, indVar, objective);

% L2 gain bound
alpha_opt = info.obj;
gamma = sqrt(alpha_opt);
disp(['L2 gain bound: ' num2str(gamma)])

% Substitute optimal values into the storage function
Vopt = subs(V, dopt);
```

```
L2 gain bound: 2.8182
```

Verify SOS constraints

```
[issos(Vopt) issos(-jacobian(Vopt,x)*f - h^2 + alpha_opt*u^2)]
```

```
ans =

     1     1
```

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