Stability Certification of a Cyclic Interconnection of LTI Systems

This example certifies stability of a cyclic interconnection of LTI systems from Ex. 1.3. The interconnection is of the form in Equation 2.13. Two equivalent sufficient conditions can certify the stability of the interconnection. **This example can be rerun, and outcomes will change, depending on the random subsystem data.**

Contents

Requirements

This script requires the CVX toolbox: http://cvxr.com/cvx/

Define the Interconnection

Number of subsystems

```
N = 5;
```

Negative feedback cyclic interconnection as in Equation 2.13

```
M = zeros(N,N);
M(1,N) = -1;
M(2:N, 1:N-1) = eye(N-1);
```

Define parameters and state space formulation for each system

Generate N random subsystems of the form in Example 1.3, with positive values for L and γ. The subsystems are formed with $k > \mu > 0$, ensuring that each is **output-strict passive**.

```
n = 2;
nUY = 1;
% Create array of subsystems, initialized with static gains of 0
Gsubsys = ss(zeros(nUY,nUY,N));

% Fill array
for j = 1:N
    % Individual system parameters
```

```
    L = rand;
    gamma = 1.3*rand;
    mu = rand;
    k = mu + rand;

    % State Space Matrices
    A = [0 1; -L -k];
    B = [0; gamma];
    C = [mu 1];
    D = 0;
    Gsubsys(:,:,j) = ss(A,B,C,D);
end
```

Analyze OSP characteristic of each Subsystem

For each subsystem, find the maximum epsilon for which the LMI which certifies output-strict passivity. This maximum is referred to as the *output-strict passivity parameter.*

```
epsVector = zeros(N,1);
for j=1:N
    [A,B,C,D] = ssdata(Gsubsys(:,:,j));
    T = [zeros(nUY, n) eye(nUY); C D];
    cvx_begin sdp quiet
        variable P(n,n) semidefinite
        variable epsilon nonnegative
        maximize epsilon
            [A'*P+P*A P*B; B'*P 0] <= T'*[zeros(nUY) 1/2*eye(nUY); 1/2*eye(nUY) -epsilon*eye(nUY)]*
T;
    cvx_end

    % Save the value of the output-strict passivity parameter, epsilon, for
    % each subsystem
    epsVector(j) = epsilon;
end
```

Certify Stability of the Interconnection

The system is stable if there exists a diagonal P > 0 such that P*(M-E)+(M-E)'*P <= 0, where E is the (block) diagonal matrix of the epsilon parameters from the output-strict passivity analysis. **Note**: code below is specific for nUY=1.

```
% Diagonal matrix of epsilon values
E = diag(epsVector);

% Formulate and solve with CVX
cvx_begin sdp quiet
variable P(N,N) diagonal
minimize 0
    P >= eye(N);              % Ensure P is positive definite
    P*(M-E)+(M-E)'*P <= 0;        % Homogeneous in P
cvx_end
if strcmp(cvx_status,'Solved')
    disp('LMI: certifies stability.');
else
    disp('LMI: inconclusive.');
```

```
end
```

```
LMI: certifies stability.
```

Compare to Secant Condition

Compare the result to the equivalent condition given for stability given by Equation 2.14

```
if prod(epsVector) >= cos(pi/N)^N
    disp('Secant: certifies stability.');
else
    disp('Secant: inconclusive.');
end
```

```
Secant: certifies stability.
```

Note that the LMI is feasible if and only if the secant condition passes. *Remember though, these equivalent conditions are only sufficient conditions for stability of the cyclic interconnection.*

Directly check stability of interconnection

For comparison, create block-diagonal augmentation of subsystems, form interconnection with M, and check stability directly.

```
G = ss(zeros(N*nUY,N*nUY));
for j=1:N
    uIdx = ((j-1)*nUY+1):j*nUY;
    yIdx = uIdx;
    G(yIdx,uIdx) = Gsubsys(:,:,j);
end

if isstable(lft(M,G))
    disp('Direct calculation: stable.')
else
    disp('Direct calculation: unstable.')
end
```

```
Direct calculation: stable.
```

Conclusion

A cyclic interconnection of different subsystems, all output-strict passive, is analyzed in 3 manners: the LMI and Secant conditions, which only use the dissipativity properties (specifically the output-strict passivity parameter of each subsystem), along with the interconnection matrix; and for comparison, a direct eigenvalue calculation of the entire closed-loop "A" matrix. **Rerunning the example produces different outcomes, as the subsystem models are randomly generated.**

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