

PROJETO DE SISTEMAS DE CONTROLE VIA LMIS USANDO O MATLAB

Edvaldo Assunção,

Departamento de Engenharia Elétrica, Faculdade de Engenharia de Ilha Solteira,
Universidade Estadual Paulista - UNESP,
CP 31, 15385-000, Ilha Solteira, SP
edvaldo@dee.feis.unesp.br

Marcelo Carvalho Minhoto Teixeira,

Departamento de Engenharia Elétrica, Faculdade de Engenharia de Ilha Solteira,
Universidade Estadual Paulista - UNESP,
CP 31, 15385-000, Ilha Solteira, SP
marcelo@dee.feis.unesp.br

Resumo: Pretende-se com este curso introduzir os conceitos básicos de projeto de sistemas de controle usando LMIs. É suposto que o leitor não tem nenhum conhecimento prévio sobre desigualdades matriciais lineares, bem como sobre estabilidade de sistemas dinâmicos. Os sistemas dinâmicos abordados são lineares ou não-lineares, com ou sem incertezas paramétricas. Os projetos são implementados usando-se o software MATLAB (LMI Control Toolbox).

Palavras-chave: controle, incertezas paramétricas, LMI, modelo fuzzy, MATLAB.

1 Introdução

Segundo Boyd (Boyd, El Ghaoui, Feron & Balakrishnan 1994), a história do uso das LMIs (do inglês, *Linear Matrix Inequalities*) na análise de sistemas dinâmicos tem mais de 100 anos. A história começa no ano de 1890, quando Liapunov publicou seu trabalho introduzindo o que chamamos de teoria de Liapunov. Ele formulou o estudo de estabilidade de sistemas dinâmicos através de uma desigualdade matricial. O grande destaque se deu na década de 1940, Lur'e, Postikov e outros na extinta União Soviética, aplicaram o método de Liapunov em alguns problemas práticos de engenharia, especialmente no problema de estabilidade de sistemas de controle, com uma não-linearidade no atuador. Embora não tenham descrito o problema na forma de LMIs, na introdução do livro de Lur'e tem-se (Boyd et al. 1994): “Este livro representa a primeira demonstração de que as idéias de Liapunov, ... tem aplicações práticas”.

O próximo grande ponto na história ocorreu na década de 1960, quando Yakubovich, Popov, Kalman e outros reduziram a solução de LMIs a um critério gráfico simples que agora é chamando de *positive-real lemma*.

Em 1984, N. Karmarkar introduziu um novo método de programação linear que resolve os problemas descritos na forma de LMIs com convergência polinomial, muito eficiente na prática. Em 1988, Nesterov e Nemirovskii desenvolveram o método de pontos interiores que se aplicam diretamente nos problemas de programação convexa que envolvem LMIs.

Atualmente, as LMIs estão sendo objeto de muitas pesquisas por renomados pesquisadores mundiais, tendo sido aplicadas nas mais diversas áreas: controle de sistemas contínuos e discretos no tempo (ElGhaoui & Niculescu 2000), controle ótimo, controle robusto (VanAntwerp & Braatz 2000), redução de modelos ((Assunção 2000), (Assunção & Peres 1998c), (Assunção & Peres 1998a), (Assunção & Peres 1998b), (Assunção & Peres 1999a), (Assunção & Peres 1999b)), controle de sistemas não-lineares, teoria de filtros robustos (Palhares 1998), identificação de sistemas, controle com estrutura variável (Teixeira, Lordelo & Assunção 2000), controle usando lógica fuzzy (Teixeira, Pietrobom & Assunção 2000), (Tanaka & Wang 1988) etc.

Este trabalho tem o objetivo de ser um texto de estudo introdutório sobre o uso LMIs para resolver problemas de controle. Foi suposto que o leitor não tem nenhum conhecimento prévio sobre LMIs. Inicia-se com o estudo de estabilidade por Liapunov, descrito na forma de LMIs, passa por projeto de controladores robustos, ilustra a aplicação do complemento de Schur na obtenção de LMIs, controle \mathcal{H}_2 e \mathcal{H}_∞ , terminando com o controle de sistemas não-lineares usando lógica fuzzy. As LMIs são resolvidas usando as ferramentas disponíveis no MATLAB, o *LMI Control Toolbox* (P. Gahinet 1995), sendo este material uma introdução à programação para resolução de LMIs.

Desejamos agradecer à Professora Vilma A. Oliveira (EESC-USP) pelas valiosas sugestões sobre a estrutura deste curso.

2 Descrição de um sistema dinâmico em espaço de estados

Enquanto a teoria de controle convencional é baseada na relação entre a entrada e a saída (função de transferência), a teoria de controle moderno se baseia na descrição das equações do sistema dinâmico em termos de n equações diferenciais de primeira ordem, usando uma notação matricial (Ogata 1997), (G. F. Franklin & Emami-Naeini 1994), (Dorf & Bishop 1995). De forma geral temos:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} u \quad (1)$$

$$y = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

ou simplifcadamente,

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (2)$$

sendo u a entrada do sistema dinâmico, y a saída, $[x_1 x_2 \cdots x_n]'$ os estados internos, a_{ij} , b_i e c_i são os parâmetros do sistema. Se o sistema dinâmico tem uma entrada e uma saída, B e C são vetores. Se o sistema tem mais de uma entrada e mais de uma saída, B e C são matrizes.

Exemplo: Considere o sistema dinâmico definido pela seguinte equação diferencial

$$\ddot{y} + 6\dot{y} + 11y = 6u$$

sendo u a entrada e y a saída. As variáveis de estado podem ser escolhidas como:

$$\begin{aligned} x_1 &= y \\ x_2 &= \dot{y} \\ x_3 &= \ddot{y} \end{aligned}$$

então

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= -6x_1 - 11x_2 - 6x_3 + 6u \end{aligned}$$

que podem ser colocados na forma matricial dada em (1):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} u \quad (3)$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

ou na forma de (2) sendo:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} \quad \text{e} \quad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Esta formulação possibilita a análise e projeto de sistemas dinâmicos de controle mais abrangente que a formulação por função de transferência. Dentre estes, é possível estudar controlabilidade, observabilidade, projeto de observadores de estados, estabilidade segundo Liapunov, estudo de sistemas dinâmicos não-lineares etc. O nosso interesse neste curso é o estudo de estabilidade segundo Liapunov.

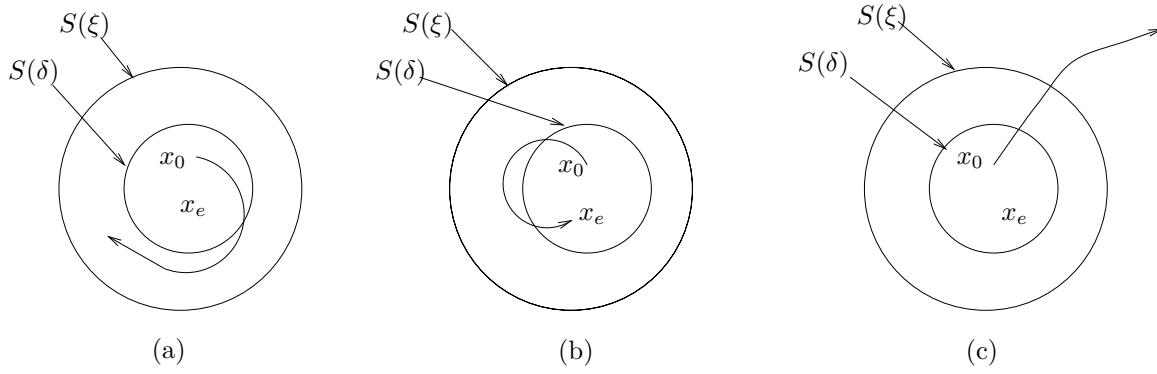


Figura 1: Interpretação de estabilidade: (a) estado de equilíbrio estável, (b) estado de equilíbrio assintoticamente estável e (c) estado de equilíbrio instável.

3 Análise de estabilidade segundo Liapunov

Considere o sistema dinâmico linear autônomo (relaxado) com $u = 0$ em (2), ou seja,

$$\dot{x} = Ax \quad (4)$$

Um estado de equilíbrio é alcançado quando $\dot{x} = 0$. Todas as derivadas nulas significa que os estados não estão variando no tempo e portanto, são indicados como estados em equilíbrio ou ponto de equilíbrio x_e .

3.1 Estabilidade no sentido de Liapunov

É necessário definir uma região esférica de raio k em torno do estado de equilíbrio x_e por

$$\|x - x_e\| \leq k$$

sendo que $\|x - x_e\|$ é chamada de norma Euclidiana e é definida por:

$$\|x - x_e\| = [(x_1 - x_{1e})^2 + (x_2 - x_{2e})^2 + \dots + (x_n - x_{ne})^2]^{\frac{1}{2}}$$

Seja $S(\delta)$ a região que consiste em todos os pontos tais que

$$\|x - x_e\| \leq \delta$$

Seja x_0 o estado inicial, e seja $S(\xi)$ a região que consiste em todos os pontos para os quais

$$\|x - x_e\| \leq \xi, \quad \forall t > 0$$

Um estado de equilíbrio x_e do sistema (4) é dito estável no sentido de Liapunov se, correspondendo a cada $S(\xi)$ houver um $S(\delta)$ tal que as trajetórias iniciadas em $S(\delta)$ não deixem $S(\xi)$ à medida que t cresce.

3.2 Estabilidade assintótica

Um estado de equilíbrio x_e do sistema (4) é dito assintoticamente estável se for estável no sentido de Liapunov e se toda solução começando em $S(\delta)$ converge para x_e , sem deixar $S(\xi)$, à medida que t aumenta.

3.3 Instabilidade

Um estado de equilíbrio x_e do sistema (2) é dito instável se, para algum número real $\xi > 0$ e todo número real $\delta > 0$, não importando quão pequeno ele seja, há sempre um estado x_0 em $S(\delta)$ tal que a trajetória, começando neste estado deixa a região $S(\xi)$.

A Figura 1 mostra a interpretação geométrica da estabilidade, estabilidade assintótica e instabilidade.

Exemplo: A Figura 2 mostra uma bola sob a ação da gravidade e exemplifica os estados de equilíbrio em uma topologia, mostrando dois pontos de equilíbrio, um estável e outro instável.

Para o estudo de estabilidade, é necessário as seguintes definições:

Funções escalares positivas definidas: uma função escalar $V(x)$ é dita positiva definida em uma região Ω se $V(x) > 0$ para todos os estados não nulos x na região Ω e $V(0) = 0$.

Figura 2: Exemplo de estado de equilíbrio estável e instável.

Funções escalares negativas definidas: uma função escalar $V(x)$ é dita negativa definida se $-V(x)$ for positiva definida.

Funções escalares positivas semidefinidas: uma função escalar $V(x)$ é dita positiva semidefinida se for positiva em todos os estados de uma região Ω , exceto na origem e em outros estados onde seu valor é zero, $V(x) \geq 0$.

Funções escalares negativas semidefinidas: uma função escalar $V(x)$ é dita negativa semidefinida se $-V(x)$ for positiva semidefinida.

Exemplo: Função escalar positiva definida:

$$V(x) = x_1^2 + 2x_2^2$$

Função escalar positiva semidefinida:

$$V(x) = (x_1 + x_2)^2$$

Função escalar negativa definida:

$$V(x) = -x_1^2 - (3x_1 + 2x_2)^2$$

Forma quadrática: a forma quadrática constitui uma classe de funções escalares que desempenham um papel importante na análise de estabilidade de sistemas dinâmicos, a forma geral é dada por:

$$V(x) = x'Px = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

A definição positiva da forma quadrática ou da forma hermitiana $V(x)$ pode ser determinada com o estudo dos autovalores de P . Se os autovalores de P forem todos positivos então pode concluir-se que P é positiva definida. No MATLAB os autovalores de uma matriz são obtidos usando a função “eig”.

Exemplo: Mostrar que a seguinte forma quadrática é positiva definida:

$$V(x) = 10x_1^2 + 4x_2^2 + x_3^2 + 2x_1x_2 - 2x_2x_3 - 4x_1x_3$$

Sol.: A forma quadrática matricial é:

$$V(x) = x'Px = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} 10 & 1 & -2 \\ 1 & 4 & -1 \\ -2 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

O seguinte programa MATLAB resolve o problema:

```
% Exemplo de verificação de positividade
clear
P=[10 1 -2;1 4 -1; -2 -1 1];
autovalores=eig(P);
if min(autovalores) > 0
    disp('Matriz positiva definida');
    disp('P= '), disp(P)
    disp('Autovalores de P = '), disp(autovalores);
end
if max(autovalores) < 0
    disp('Matriz negativa definida');
    disp('P= '), disp(P)
    disp('Autovalores de P = '), disp(autovalores);
end
```

O resultado é o seguinte:

Matriz positiva definida

P=

$$\begin{bmatrix} 10 & 1 & -2 \\ 1 & 4 & -1 \\ -2 & -1 & 1 \end{bmatrix}$$

Autovalores de P =

0.4037

3.9581

10.6382

3.4 Análise de estabilidade segundo Liapunov

Considere o sistema linear autônomo dado em (4) e reproduzido a seguir:

$$\dot{x} = Ax$$

sendo x o vetor de estados e A uma matriz constante. Se A é não-singular, então o único estado de equilíbrio é a origem $x = 0$.

Para este sistema defina uma função de Liapunov:

$$V(x) = x'Px > 0 \quad (\text{positiva definida})$$

A derivada em relação ao tempo é:

$$\dot{V}(x) = \dot{x}'Px + x'P\dot{x} = (Ax)'Px + x'PAx = x'(A'P + PA)x$$

Como $V(x)$ foi escolhida sendo definida positiva, para se ter estabilidade assintótica é necessário que $\dot{V}(x)$ seja definida negativa:

$$\dot{V}(x) < 0 \quad (\text{negativa definida})$$

então é necessário para a estabilidade de (4) que

$$A'P + PA < 0$$

Teorema 1: Considere o sistema autônomo descrito pela equação (4):

$$\dot{x} = Ax$$

sendo que x é um vetor de estado e A é uma matriz $n \times n$ constante e não-singular. Uma condição necessária e suficiente para que o estado de equilíbrio $x = 0$ seja assintoticamente estável é que, exista uma matriz P positiva definida, simétrica, tal que

$$A'P + PA < 0 \quad (5)$$

Interpretação geométrica: na Figura 3 mostramos uma função $V(x)$ e o caminho da evolução dos estados do sistema relaxado (4), sendo que os estados estão restritos ao lugar geométrico delineado por $V(x)$, ou seja pela função de Liapunov escolhida. Note que $\dot{V}(x)$ é sempre negativa ao longo de $V(x)$, mostrando que $V(x)$ decresce ao longo do tempo, o que implica que o sistema é assintoticamente estável.

4 Estudo de estabilidade usando LMI

Uma importante ferramenta matemática para o estudo de sistemas descritos em espaço de estados é o uso de desigualdades matriciais lineares LMIs (do inglês: *Linear Matrix Inequalities*) (Boyd et al. 1994). Um exemplo de LMI é o estudo da positividade de uma matriz com elementos lineares:

$$P > 0$$

sendo $P = P'$.

O estudo de estabilidade usando LMI advém da equação de Liapunov. Seja a desigualdade matricial de Liapunov, equação (5):

$$A'P + PA < 0$$

para verificar se (4) é estável é necessário e suficiente verificar se as LMIs abaixo são factíveis:

$$\begin{array}{rcl} A'P + PA & < & 0 \\ P & > & 0 \end{array}$$

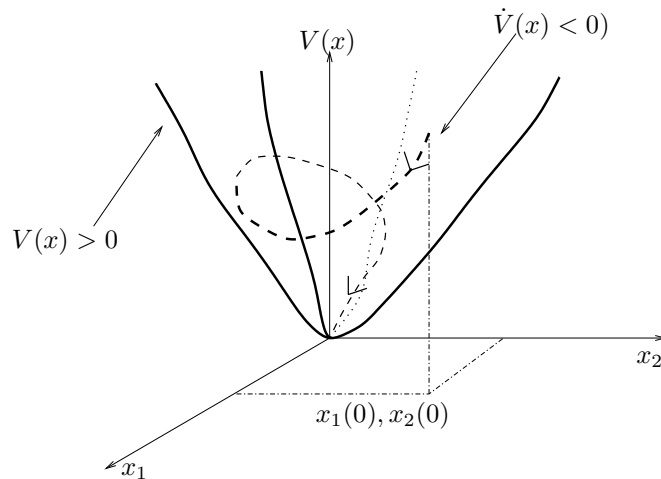


Figura 3: Interpretação geométrica do teorema.

com $P = P'$.

Exemplo: Estude a estabilidade do sistema dado pela equação (3) usando LMI.

Sol.: O seguinte programa MATLAB resolve o problema:

```
% Exemplo de estudo de estabilidade usando LMIs
clear
A=[0 1 0; 0 0 1; -6 -11 -6 ];
setlmis([]); % Início da montagem das LMIs
P=lmivar(1,[length(A) 1]); % Declaração que P é uma matriz simétrica
lmiterm([1 1 1 P],1,A,'s'); % LMI #1: P*A+A'*P<0
lmiterm([-2 1 1 P],1,1); % LMI #2: P > 0
lmiestabilidade=getlmis; % Término da montagem das LMIs
[tmin,xfeasp]=feasp(lmiestabilidade); % Teste da factibilidade da lmi.
                                     %Se tmin<0: a LMI é factível

if tmin<0
    Pf=dec2mat(lmiestabilidade,xfeasp,P);
    disp('Sistema estável');
    disp('P= '), disp(Pf)
    disp('Autovalores de P'), disp(eig(Pf))
    disp('Autovalores de A'), disp(eig(A))
else
    disp('Sistema instável')
end
```

O resultado é o seguinte:

Solver for LMI feasibility problems $L(x) < R(x)$
 This solver minimizes t subject to $L(x) < R(x) + t \cdot I$
 The best value of t should be negative for feasibility

Iteration : Best value of t so far

1 -0.032822

Result: best value of t : -0.032822

Sistema estável P=

1.0615	0.3416	0.0547
0.3416	1.1318	0.0766
0.0547	0.0766	0.0693

Autovalores de P

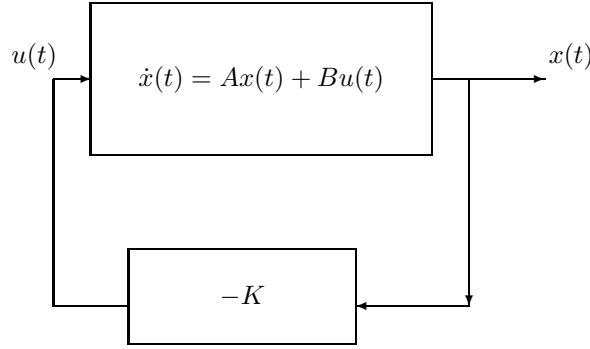


Figura 4: Sistema realimentado.

```
0.0628
0.7534
1.4464
```

Autovalores de A

```
-1.0000
-2.0000
-3.0000
```

Para que este programa seja executado, é necessário que no MATLAB esteja instalado o *LMI Control Toolbox* (P. Gahinet 1995). Pode-se utilizar o software LMISol de domínio público (de Oliveira, Farias & Geromel 1997), porém todos os programas mostrados neste curso foram feitos para o MATLAB.

Note que somente a investigação dos autovalores de A já seria suficiente para determinar a estabilidade de (3). A formulação via LMI possibilita o projeto de controladores para sistemas com ou sem incertezas, de uma forma simples, isto será abordado mais adiante.

A montagem das LMIs podem ser facilmente realizada usando o editor gráfico de LMIs que o MATLAB possui, veja o Apêndice deste trabalho.

5 Projeto de controladores usando LMI

Dada a planta (2), projete a matriz K do controlador:

$$u = -Kx \quad (6)$$

para que estabilize o sistema controlado mostrado na Figura 4.

Substituindo (6) em (2) temos:

$$\dot{x} = Ax - BKx = (A - BK)x \quad (7)$$

Neste caso, o estudo de estabilidade segundo Liapunov para este sistema realimentado é realizado através do estudo das seguintes desigualdades matriciais:

$$\begin{array}{rcl} P(A - BK) + (A - BK)'P & < & 0 \\ P & > & 0 \end{array} \quad (8)$$

sendo $P = P'$. Portanto, deve-se resolver (8) obtendo-se o valor de K . Note que (8) não é uma LMI pois tem o termo PBK . Assim, é necessário a seguinte manipulação de (8):

$$PA - PBK + A'P - K'B'P < 0 \quad (9)$$

multiplicando-se ambos os lados de (9) por P^{-1} , $P > 0$, tem-se

$$P^{-1}[PA - PBK + A'P - K'B'P]P^{-1} < 0$$

então

$$AP^{-1} - BKP^{-1} + P^{-1}A' - P^{-1}K'B' < 0$$

defina $X = P^{-1}$ e $G = KP^{-1} = KX$, temos

$$\begin{array}{ccc} AX - BG + XA' - G'B' & < & 0 \\ X & > & 0 \end{array} \quad (10)$$

sendo $X = X'$. Note que P^{-1} existe pois $P > 0$, ou seja todos autovalores de P são diferentes de zero ou melhor, maiores que zero. O controlador é dado por: $K = GX^{-1}$. Note que com (10) o problema (8) ficou descrito na forma de LMIs.

Exemplo: Usando a realimentação de estados, projete um controlador que estabilize a seguinte planta:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Sol.: O seguinte programa MATLAB resolve o problema.

```
% Exemplo de projeto de controladores usando LMIs
clear
A=[1 0 0;0 -1 0;0 0 4];
B=[1 1 1]';
n=length(A);
setlmis([]); % Início da montagem das LMIs
X=lmivar(1,[n 1]); % Declaração que X é uma matriz simétrica nxn
G=lmivar(2,[1 n]); % Declaração que G é uma matriz retangular 1xn
lmiterm([1 1 1 X],A,1,'s'); % LMI #1: X*A'+A*X
lmiterm([1 1 1 G],B,-1,'s'); % LMI #1: -G*B'-B*G
lmiterm([-2 1 1 X],1,1); % LMI #2: X > 0
lmicontrolador=getlmis; % Término da montagem das LMIs
[tmin,xfesp]=feasp(lmicontrolador); % Teste da factibilidade da lmi.
%Se tmin<0: a LMI é factível

if tmin<0
    Xf=dec2mat(lmicontrolador,xfesp,X);
    Gf=dec2mat(lmicontrolador,xfesp,G);
    K=Gf*inv(Xf);
    disp('Controlador'), disp(K)
else
    disp('Sistema não controlável')
end
```

6 Projeto de controladores robustos usando LMI

Se a planta possui incertezas politópicas, o projeto do controlador robusto através de LMIs é feito utilizando-se o conceito de estabilidade quadrática.

Definição de estabilidade quadrática: um sistema incerto com incertezas do tipo politópicas, com m vértices, é quadraticamente estabilizável se existe X e G tais que as seguintes LMIs sejam factíveis em conjunto:

$$\begin{array}{ccc} A_i X - BG + XA'_i - G'B' & < & 0 \\ X & > & 0 \end{array} \quad (11)$$

sendo A_i o i -ésimo vértice do politopo, $i = 1, 2, \dots, m$. Mais detalhadamente, o projeto do controlador robusto que estabiliza o sistema incerto consiste em resolver as seguintes LMIs simultaneamente:

$$\begin{array}{ccc} A_1 X - BG + XA'_1 - G'B' & < & 0 \\ A_2 X - BG + XA'_2 - G'B' & < & 0 \\ \vdots & & \\ A_m X - BG + XA'_m - G'B' & < & 0 \\ X & > & 0 \end{array}$$

sendo $X = X'$ e m o número total de vértices do politopo. O controlador é dado por: $K = GX^{-1}$.

Exemplo: Projete um controlador robusto que estabilize a seguinte planta incerta, usando a realimentação de estados:

$$A = \begin{bmatrix} a & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & b \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

sendo

$$0.5 < a < 1.5$$

e

$$2 < b < 6$$

Sol.: Neste caso, os vértices do politopo são:

$$A_1 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 6 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 6 \end{bmatrix},$$

O seguinte programa MATLAB resolve o problema.

```
% Exemplo de projeto de controladores robustos usando LMIs
clear
A1=[0.5 0 0;0 -1 0;0 0 2];
A2=[1.5 0 0;0 -1 0;0 0 2];
A3=[0.5 0 0;0 -1 0;0 0 6];
A4=[1.5 0 0;0 -1 0;0 0 6];
B=[1 1 1]';
n=length(A1);
setlmis([]); % Início da montagem das LMIs
X=lmivar(1,[n 1]);% Declaração que X é uma matriz simétrica nxn
G=lmivar(2,[1 n]);% Declaração que G é uma matriz retangular 1xn
lmiterm([1 1 1 X],A1,1,'s'); % LMI #1: X*A1'+A1*X
lmiterm([1 1 1 G],B,-1,'s'); % LMI #1: -G*B'-B*G
lmiterm([2 1 1 X],A2,1,'s'); % LMI #2: X*A2'+A2*X
lmiterm([2 1 1 G],B,-1,'s'); % LMI #2: -G*B'-B*G
lmiterm([3 1 1 X],A3,1,'s'); % LMI #3: X*A3'+A3*X
lmiterm([3 1 1 G],B,-1,'s'); % LMI #3: -G*B'-B*G
lmiterm([4 1 1 X],A4,1,'s'); % LMI #4: X*A4'+A4*X
lmiterm([4 1 1 G],B,-1,'s'); % LMI #4: -G*B'-B*G
lmiterm([-5 1 1 X],1,1); % LMI #5: X > 0
lmicontrolador=getlmis; % Término da montagem das LMIs
[tmin,xfeasp]=feasp(lmicontrolador); % Teste da factibilidade da lmi.
%Se tmin<0: a LMI é factível

if tmin<0
    Xf=dec2mat(lmicontrolador,xfeasp,X);
    Gf=dec2mat(lmicontrolador,xfeasp,G);
    K=Gf*inv(Xf);
    disp('Controlador'), disp(K)
else
    disp('Sistema não controlável')
end
```

Pode-se incluir neste projeto restrições de taxa de decaimento, restrições na amplitude da entrada e da saída, vide (Teixeira, Pietrobom & Assunção 2000).

7 Complemento de Schur

O estudo das propriedades das LMIs podem ser encontradas em (VanAntwerp & Braatz 2000), (Boyd et al. 1994) e (ElGhaui & Niculescu 2000). Um importante resultado sobre LMI é o complemento de Schur. O complemento de Schur converte uma classe de desigualdades matriciais não-lineares, que aparecem regularmente na teoria de controle, em desigualdades matriciais lineares (LMIs). O complemento de Schur é ilustrado a seguir.

Seja a desigualdade matricial não-linear:

$$Q(x) - S(x)R(x)^{-1}S(x)' > 0, \quad R(x) > 0 \quad (12)$$

sendo $Q(x) = Q(x)'$, $R(x) = R(x)'$ e $S(x)$ são afins em x . Então o sistema (12) é equivalente a

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)' & R(x) \end{bmatrix} > 0$$

Uma demonstração do complemento de Schur pode ser encontrada em (VanAntwerp & Braatz 2000).

Um exemplo de aplicação do complemento de Schur está no cálculo da norma \mathcal{H}_∞ de um sistema linear via LMI.

8 Cálculo da norma \mathcal{H}_∞ de um sistema dinâmico linear usando LMI

Considere o sistema dinâmico dado em (2), cuja função de transferência é

$$H(s) = C(sI - A)^{-1}B$$

A norma \mathcal{H}_∞ deste sistema é dada pelo seguinte problema de otimização:

$$\begin{aligned} \|H\|_\infty = \min \quad & \gamma \\ \text{s.a.} \quad & A'P + PA + \gamma^{-2}PBB'P + C'C < 0 \\ & P > 0 \\ & \gamma > 0 \end{aligned} \quad (13)$$

sendo $P = P'$. Uma demonstração de (13) é mostrada em (Boyd et al. 1994).

Aplicando o complemento de Schur no sentido inverso em

$$-A'P - PA - \gamma^{-2}PBB'P - C'C > 0$$

adotando:

$$Q(x) = -A'P - PA - C'C$$

$$S(x) = -PB$$

e

$$R(x) = \gamma^2$$

tem-se

$$\begin{bmatrix} -A'P - PA - CC' & -PB \\ -B'P & \gamma^2 \end{bmatrix} > 0$$

ainda

$$\begin{bmatrix} A'P + PA + CC' & PB \\ B'P & -\gamma^2 \end{bmatrix} < 0$$

adotando $\mu = \gamma^2$, o problema de otimização (13) tem a seguinte forma equivalente:

$$\begin{aligned} \|H\|_\infty = \min \quad & \mu \\ \text{s.a.} \quad & \begin{bmatrix} A'P + PA + CC' & PB \\ B'P & -\mu \end{bmatrix} < 0 \\ & P > 0 \\ & \mu > 0 \end{aligned} \quad (14)$$

sendo $P = P'$.

Exemplo: Determine a norma \mathcal{H}_∞ do seguinte sistema dinâmico:

$$A = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \quad \text{e} \quad C = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

O seguinte programa MATLAB resolve o problema.

```
% Determinação da norma Hoo via LMIs
clear
A=[-1 0 1;0 -2 0;0 0 -3];
B=[1 0 2]';
C=[1 1 0];
n=length(A);
setlmi([]); % Início da montagem das LMIs
mi=lmiavar(1,[1 1]); % Declaração que mi é uma variável escalar
P=lmiavar(1,[n n]); % Declaração que P é uma matriz simétrica
lmiterm([1 1 1 P],A',1,'s'); % LMI #1: A'*P+P*A
lmiterm([1 1 1 0],C'*C); % LMI #1: C'*C
lmiterm([1 2 1 P],B',1); % LMI #1: B'*P
lmiterm([1 2 2 mi],-1,1); % LMI #1: mi
lmiterm([-2 1 1 P],1,1); % LMI #2: P>0
lmiterm([-3 1 1 mi],1,1); % LMI #3: mi>0
lmiinfy=getlmi; % Término da montagem das LMIs
nvar1=decnbr(lmiinfy);
objc1=zeros(nvar1,1); % Definindo a variável a ser otimizada
objc1(1)=1; % Variável de otimização: mi
[objcopt,xopt]=mincx(lmiinfy,objc1); % Resolvendo o problema de
miopt=dec2mat(lmiinfy,xopt,mi); % otimização
Hoo=sqrt(miopt);
disp('Norma Hoo'), disp(Hoo)
```

9 Projeto de controlador ótimo \mathcal{H}_2 usando LMI

Considere o sistema dinâmico dado pela equação

$$\begin{aligned} \dot{x} &= Ax + B_1 w + B_2 u \\ y &= Cx \end{aligned} \quad (15)$$

sendo w a entrada exógena do sistema (ruído).

Suponha que o controle do sistema está sendo feito com a realimentação dos estados:

$$u = Kx$$

Deseja-se obter K de tal forma que a influência do ruído na saída y do sistema seja o menor possível. Portanto, deseja-se minimizar a norma \mathcal{H}_2 entre a entrada exógena w e a saída de interesse y . O projeto do controlador K de tal forma a minimizar a norma \mathcal{H}_2 deste sistema é dado pelo seguinte problema de otimização:

$$\begin{aligned} \min \quad & Tr(Z) \\ \text{s.a.} \quad & \begin{bmatrix} X & B_1 \\ B_1' & Z \end{bmatrix} > 0 \\ & \begin{bmatrix} AX + B_2 Y + XA' + Y'B_2' & XC' \\ CX & -I \end{bmatrix} < 0 \\ & X > 0 \\ & Z > 0 \end{aligned} \quad (16)$$

sendo $X = X'$, $Z = Z'$. Após realizar a otimização acima, o controlador K é obtido com: $K = YX^{-1}$, sendo X e Y soluções ótimas de (16).

Problema: Implemente um programa MATLAB que realize o projeto acima para o seguinte sistema:

$$A = \begin{bmatrix} -2 & 1 & 1 & 1 \\ 3 & 0 & 0 & 2 \\ -1 & 0 & -2 & -3 \\ -2 & -1 & 2 & -1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{e} \quad C = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

10 Controle com lógica fuzzy de sistemas não-lineares

Os modelos fuzzy Takagi-Sugeno (TSK) têm sido úteis na descrição aproximada de sistemas não-lineares. É bem conhecido que sistemas fuzzy aditivos $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^p$, como o modelo fuzzy TSK, podem aproximar uniformemente qualquer função contínua $\mathbf{f} : \mathbf{U} \subset \mathbb{R}^n \rightarrow \mathbb{R}^p$ em um domínio compacto (fechado limitado) \mathbf{U} (Kosko 1997). No caso do modelo fuzzy TSK, quanto maior o número de modelos locais, melhor será a aderência do modelo à equação diferencial não-linear da planta.

O exemplo abaixo ilustra a aproximação obtida pelos modelos fuzzy TSK.

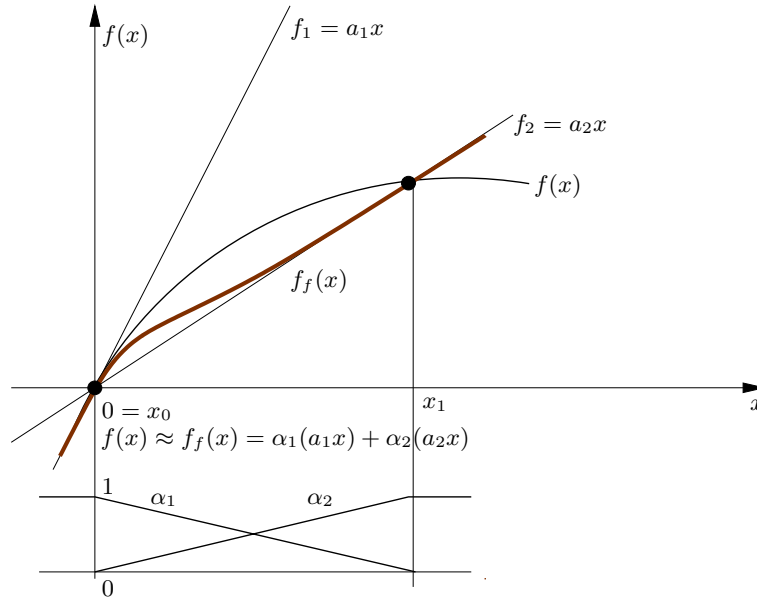


Figura 5: Ilustração da aproximação obtida por modelos fuzzy TSK.

Considere a função não-linear $f(x)$ descrita na Figura 5. Note que esta função pode ser aproximada, para $x \approx x_0 = 0$, por $f_1(x) = a_1x$, que é a reta tangente desta curva em $x = 0$. Uma aproximação linear para esta função, para $x \approx x_1$, é $f_2(x) = a_2x$; observe que esta segunda aproximação linear não é tão boa quanto a primeira aproximação linear, pois $f_2(x_0) = 0$ e $f_2(x_1) \neq 0$. Adotando-se $f_1(x)$ e $f_2(x)$ como modelos locais, e as funções $\alpha_1(x)$, $\alpha_2(x)$ definidas na Figura 5 (observe que $\alpha_1(x) + \alpha_2(x) = 1$), um modelo fuzzy TSK para $f(x)$ seria $f_f(x) = \alpha_1(x)f_1(x) + \alpha_2(x)f_2(x)$, como ilustrado na Figura 5. Pode-se observar que para $x \approx x_0$, então $\alpha_1 \approx 1$, $\alpha_2 \approx 0$ e $f_f(x) \approx f_1(x)$ e para $x \approx x_1$, então $\alpha_2 \approx 1$, $\alpha_1 \approx 0$ e $f_f(x) \approx f_2(x)$. Finalmente, verifique que $f_f(x)$ proporciona uma aproximação da função $f(x)$ muito melhor do que as funções $f_1(x)$ (linearização em torno de um ponto de operação) ou $f_2(x)$, por exemplo para $x_0 \leq x \leq x_1$. Obviamente, se aumentarmos o número de modelos locais, a aproximação torna-se melhor. Esse exemplo simples mostra o potencial dos modelos fuzzy TSK, no tratamento de funções e/ou de sistemas não-lineares.

10.1 Modelo Fuzzy Takagi-Sugeno

No método de projeto TSK, uma dada planta não-linear é representada pelo modelo fuzzy Takagi-Sugeno (Takagi & Sugeno 1985). Este método de modelagem fuzzy é simples. As dinâmicas do sistema são capturadas por um conjunto de implicações fuzzy que caracterizam relações locais no espaço de estados. A principal característica de um modelo Takagi-Sugeno é a descrição das dinâmicas locais de cada implicação fuzzy (regra) por um modelo de sistema linear. O modelo fuzzy global do sistema é obtido pela combinação fuzzy dos modelos do sistema linear.

Especificamente, o sistema fuzzy Takagi-Sugeno é descrito pelas regras fuzzy SE-ENTÃO, que representam localmente relações lineares entre a entrada e a saída de um sistema.

A descrição local da planta dinâmica a ser controlada está disponível nos termos dos modelos lineares locais. Portanto tem-se os seguintes modelos lineares locais:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}_i \mathbf{x}(t); \end{aligned} \quad (17)$$

sendo $i = 1, 2, \dots, r$ (r é o número de modelos lineares), o vetor estado $\mathbf{x}(t) \in \mathbb{R}^n$, o vetor entrada $\mathbf{u}(t) \in \mathbb{R}^m$, o vetor saída $\mathbf{y}(t) \in \mathbb{R}^q$, $\mathbf{A}_i \in \mathbb{R}^{n \times n}$, $\mathbf{B}_i \in \mathbb{R}^{n \times m}$ e $\mathbf{C}_i \in \mathbb{R}^{q \times n}$. A informação acima é então fundida com as regras SE-ENTÃO disponíveis, onde a i -ésima regra pode ter a forma:

$$\begin{aligned} \text{Regra } i &: \text{ SE } z_1(t) \text{ é } \mathcal{M}_1^i \text{ E } \dots \text{ E } z_p(t) \text{ é } \mathcal{M}_p^i, \\ &\text{ ENTÃO } \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}_i \mathbf{x}(t); \end{cases} \end{aligned} \quad (18)$$

10.2 Reguladores com Modelos Fuzzy Takagi-Sugeno

Pode-se utilizar o conceito de Compensação Distribuída Paralela (CDP) para projetar reguladores fuzzy para estabilizar sistemas não-lineares descritos por modelos fuzzy. A idéia é projetar um compensador para cada regra do modelo fuzzy. Para cada regra, são utilizadas técnicas de projeto de controle linear. O regulador fuzzy global resultante, que é não-linear em geral, é uma combinação fuzzy de cada regulador linear individual. A CDP oferece um procedimento para projetar um regulador para o modelo fuzzy Takagi-Sugeno, onde cada regra de controle é projetada a partir da correspondente regra de um modelo Takagi-Sugeno da planta. O regulador fuzzy projetado compartilha os mesmos conjuntos de regras com o modelo fuzzy nas partes premissas. Para os modelos fuzzy (18), sendo $i = 1, 2, \dots, r$, os reguladores fuzzy via CDP possuem a seguinte estrutura:

$$\begin{aligned} \text{Regra } i &: \text{ SE } z_1(t) \text{ é } \mathcal{M}_1^i \text{ E } \dots \text{ E } z_p(t) \text{ é } \mathcal{M}_p^i, \\ &\text{ ENTÃO } \mathbf{u}(t) = -\mathbf{F}_i \mathbf{x}(t). \end{aligned} \quad (19)$$

Portanto, o regulador fuzzy é dado por

$$\begin{aligned} \mathbf{u}(t) &= -\frac{\sum_{i=1}^r w^i(\mathbf{z}(t)) \mathbf{F}_i \mathbf{x}(t)}{\sum_{i=1}^r w^i(\mathbf{z}(t))} \\ &= -\sum_{i=1}^r \alpha_i(\mathbf{z}(t)) \mathbf{F}_i \mathbf{x}(t) \\ &= -\mathbf{F}(\alpha) \mathbf{x}(t). \end{aligned} \quad (20)$$

O objetivo do projeto do regulador fuzzy é determinar os ganhos de realimentação locais \mathbf{F}_i nas partes conseqüentes. Para a lei de controle (20), as equações (17) são dadas por:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \sum_{i=1}^r \sum_{j=1}^r \alpha_i(\mathbf{z}(t)) \alpha_j(\mathbf{z}(t)) \{ \mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j \} \mathbf{x}(t) \\ &= \sum_{i=1}^r \alpha_i^2(\mathbf{z}(t)) \mathbf{G}_{ii} \mathbf{x}(t) + 2 \sum_{i < j}^r \alpha_i(\mathbf{z}(t)) \alpha_j(\mathbf{z}(t)) \left\{ \frac{\mathbf{G}_{ij} + \mathbf{G}_{ji}}{2} \right\} \mathbf{x}(t); \end{aligned} \quad (21)$$

sendo

$$\mathbf{G}_{ij} = \mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j.$$

10.3 Condições para a Estabilidade de Reguladores Fuzzy

As condições suficientes para a estabilidade de sistemas fuzzy são derivadas usando funções Lyapunov quadráticas do tipo $V(\mathbf{x}(t)) = \mathbf{x}^T(t) \mathbf{P} \mathbf{x}(t)$ (Teixeira, Pietrobom & Assunção 2000). Os principais resultados estão descritos abaixo.

Lema 1 *O ponto de equilíbrio $\mathbf{x} = 0$ do sistema fuzzy contínuo descrito por (17) é assintoticamente estável globalmente se existe uma matriz simétrica positiva definida comum \mathbf{P} tal que*

$$\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i < 0$$

para $i = 1, 2, \dots, r$; isto é, para todos os subsistemas.

Prova: Segue diretamente da aplicação da função de Lyapunov

$$V(\mathbf{x}(t)) = \mathbf{x}^T(t) \mathbf{P} \mathbf{x}(t)$$

Para a apresentação dos resultados para a estabilidade do sistema forçado, será utilizada a definição de número de regras ativas: os sistemas fuzzy (1) possuem r regras e para um certo $z(t) = z_0$ o número de regras ativas é igual ao número de termos $\alpha_1(z_0), \dots, \alpha_r(z_0)$ não nulos. Por exemplo, na Figura 6 as regras ativas para $z_1(t) \leq a$, $z_1(t) \in (a, b)$, $z_1(t) = b$, $z_1(t) \in (b, c)$, $z_1(t) = c$, $z_1(t) \in (c, d)$, $z_1(t) \geq d$ são respectivamente as regras 1, 1 e 2, 2, 2 e 3, 3, 3 e 4, 4. Assim o número de regras ativas é menor ou igual a dois.

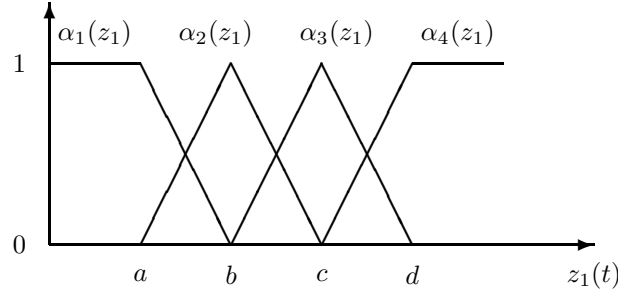


Figura 6: Exemplo de um conjunto de 4 regras fuzzy: $\alpha_1(z_1(t)), \alpha_2(z_1(t)), \alpha_3(z_1(t)), \alpha_4(z_1(t)) \in [0, 1]$ e $\alpha_1(z_1(t)) + \alpha_2(z_1(t)) + \alpha_3(z_1(t)) + \alpha_4(z_1(t)) = 1$.

Observe que na Figura 6,

$$\alpha_1(z_1(t))\alpha_3(z_1(t)) = \alpha_1(z_1(t))\alpha_4(z_1(t)) = \alpha_2(z_1(t))\alpha_4(z_1(t)) = 0, \quad \forall z_1(t) \in \mathbb{R}$$

Lema 2 O ponto de equilíbrio $\mathbf{x} = 0$ do sistema de controle fuzzy contínuo descrito por (21) é assintoticamente estável globalmente se existe uma matriz simétrica positiva definida comum \mathbf{P} tal que

$$\mathbf{G}_{ii}^T \mathbf{P} + \mathbf{P} \mathbf{G}_{ii} < 0$$

para todo $i = 1, \dots, r$ e

$$\left(\frac{\mathbf{G}_{ij} + \mathbf{G}_{ji}}{2} \right)^T \mathbf{P} + \mathbf{P} \left(\frac{\mathbf{G}_{ij} + \mathbf{G}_{ji}}{2} \right) \leq 0, \quad i < j$$

para todo $i, j = 1, \dots, r$ excetuando-se os pares (i, j) tais que $\alpha_i(\mathbf{z}(t))\alpha_j(\mathbf{z}(t)) = 0, \forall \mathbf{z}(t)$.

Prova: Segue diretamente do Lema 1.

Essas condições para estabilidade, em geral, são conservativas. Foram propostas em (Tanaka & Wang 1988) condições mais relaxadas que as condições descritas anteriormente. Recentemente as condições para a estabilidade foram flexibilizadas, vide (Teixeira, Pietrobom & Assunção 2000).

Pode-se incluir no projeto de controladores com lógica fuzzy restrições de tempo de estabelecimento, porcentagem de overshoot, limitante do sinal de entrada e limitante do sinal de saída, para maiores detalhes, vide (Teixeira, Pietrobom & Assunção 2000).

Exemplo: Projete um controlador com lógica fuzzy, de tal forma a estabilizar o sistema não-linear que possui os seguintes modelos linearizados:

$$A_1 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 6 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Sol. Implemente o seguinte programa:

```

% Projeto do controlador FUZZY via LMIs
clear
A1=[0.5 0 0 ; 0 -1 0; 0 0 2];
A2=[1.5 0 0; 0 -1 0; 0 0 6];
B1=[1 1 1]';
B2=[1 1 1]';
n=length(A1);
setlmis([]);
X=lmivar(1,[n 1]);
Y=lmivar(2,[1 n]);
Q=lmivar(2,[1 n]);
lmiterm([1 1 1 X],1,A1','s'); % LMI #1: X*A1'+A1*X
lmiterm([1 1 1 -Y],.5*1,-B1','s'); % LMI #1: -Y'*B1'
lmiterm([1 1 1 Y],.5*B1,-1,'s'); % LMI #1: -B1*Y
lmiterm([2 1 1 X],1,A2','s'); % LMI #2: X*A2'+A2*X
lmiterm([2 1 1 -Q],.5*1,-B2','s'); % LMI #2: -Q'*B2'
lmiterm([2 1 1 Q],.5*B2,-1,'s'); % LMI #2: -B2*Q
lmiterm([3 1 1 X],1,A1','s'); % LMI #3: X*A1'+A1*X
lmiterm([3 1 1 -Q],.5*1,-B1','s'); % LMI #3: -Q'*B1'
lmiterm([3 1 1 X],1,A2','s'); % LMI #3: X*A2'+A2*X
lmiterm([3 1 1 -Y],.5*1,-B2','s'); % LMI #3: -Y'*B2'
lmiterm([3 1 1 Q],.5*B1,-1,'s'); % LMI #3: -B1*Q
lmiterm([3 1 1 Y],.5*B2,-1,'s'); % LMI #3: -B2*Y
lmiterm([-4 1 1 X],1,1); % LMI #4: X
lmifuzzy=getlmis;
[tmin,xfeasp]=feasp(lmifuzzy); % Teste da factibilidade da lmi.
%Se tmin<0: a LMI é factível

if tmin<0
    Xf=dec2mat(lmifuzzy,xfeasp,X);
    Yf=dec2mat(lmifuzzy,xfeasp,Y);
    Qf=dec2mat(lmifuzzy,xfeasp,Q);
    F1=Yf*inv(Xf);
    F2=Qf*inv(Xf);
    format short;
    disp('Os controladores Fuzzy são:');disp('F1= ');disp(F1)
    disp('F2= ');disp(F2)
else
    disp('Sistema infactível')
end

```

Sendo que $X = P^{-1}$, $Y = F_1 X$ e $Q = F_2 X$. Assim os controladores Fuzzy são obtidos com:

$$F_1 = Y X^{-1}$$

$$F_2 = Q X^{-1}$$

Os resultados obtidos com a execução do programa são:

```

Solver for LMI feasibility problems L(x) < R(x)
This solver minimizes t subject to L(x) < R(x) + t*I
The best value of t should be negative for feasibility

```

```

Iteration : Best value of t so far

```

```

1          0.062809
2          0.036995
3         -0.883934

```

```

Result: best value of t: -0.883934
f-radius saturation: 0.000% of R = 1.00e+009

```

Os controladores Fuzzy são:

F1=

-7.1418 -1.0135 13.4143

F2=

-19.2520 -1.5223 34.8957

11 Conclusões

O uso de LMIs para projeto de controladores está sendo intenso, devido ao fato de possibilitar que problemas que originariamente não estejam descritos na forma linear, possam ser representados na forma de LMIs através de manipulações algébricas (complemento de Schur). Uma vez descrito na forma de LMIs, pode-se facilmente resolver o problema usando programação convexa, já implementadas no MATLAB. Ainda, a formulação na forma de LMIs possibilita o tratamento de sistemas com incertezas politópicas e sistemas não-lineares.

Referências

- Assunção, E. (2000). *Redução \mathcal{H}_2 e \mathcal{H}_∞ de Modelos Através de Desigualdades Matriciais Lineares: Otimização Local e Global*, Tese de Doutorado, UNICAMP, Campinas, SP.
- Assunção, E. & Peres, P. L. D. (1998a). Redução de modelos contínuos com critério \mathcal{H}_2 através de desigualdades matriciais lineares, *XII Congresso Brasileiro de Automática*, Vol. 3, Uberlândia, MG, pp. 879–884.
- Assunção, E. & Peres, P. L. D. (1998b). Redução de modelos discretos com critério \mathcal{H}_2 através de desigualdades matriciais lineares, *VIII Latin American Congress on Automatic Control*, Vol. 1, Viña del Mar, Chile, pp. 61–66.
- Assunção, E. & Peres, P. L. D. (1998c). Redução de modelos com critério \mathcal{H}_∞ através de desigualdades matriciais lineares: casos contínuo e discreto no tempo, *XII Congresso Brasileiro de Automática*, Vol. 3, Uberlândia, MG, pp. 885–890.
- Assunção, E. & Peres, P. L. D. (1999a). A global optimization approach for the \mathcal{H}_2 -norm model reduction problem, *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, AZ, USA, pp. 1857–1862.
- Assunção, E. & Peres, P. L. D. (1999b). A \mathcal{H}_2 and/or \mathcal{H}_∞ -norm model reduction of uncertain discrete-time systems, *Proceedings of the 1999 American Control Conference*, San Diego, CA, USA, pp. 4466–4470.
- Boyd, S., El Ghaoui, L., Feron, E. & Balakrishnan, V. (1994). *Linear Matrix Inequalities in Systems and Control Theory*, SIAM Studies in Applied Mathematics, USA.
- de Oliveira, M. C., Farias, D. P. & Geromel, J. C. (1997). *LMISol, User's guide*, UNICAMP, Campinas-SP, Brasil.
- Dorf, R. C. & Bishop, R. H. (1995). *Modern Control Systems*, Addison-Wesley Publishing Company, Reading, Massachusetts.
- G. F. Franklin, J. D. P. & Emami-Naeini, A. (1994). *Feedback Control of Dynamic Systems*, Addison Wesley, USA.
- Kosko, B. (1997). *Fuzzy Engineering*, Upper Saddle River, New Jersey 07458: Prentice-Hall, USA.
- L, ElGhaoui. & Niculescu, S. (2000). *Advances in Linear Matrix Inequality Methods in Control*, Siam, USA.
- Ogata, K. (1997). *Modern Control Engineering*, Prentice Hall, New Jersey-USA.
- P. Gahinet, A. Nemirovski, A. J. Laub e. M. Chilali. (1995). *LMI Control Toolbox User's Guide*, The Mathworks Inc., Natick, MA, USA.
- Palhares, R. M. (1998). *Filtragem Robusta: Uma Abordagem por Desigualdades Matriciais Lineares*, Tese de Doutorado, UNICAMP, Campinas, SP.
- Takagi, T. & Sugeno, M. (1985). Fuzzy Identification of Systems and Its Applications to Modeling and Control, *IEEE Transactions on Systems, Man, and Cybernetics* **15**(1): 116–132.
- Tanaka, K., I. T. & Wang, H. O. (1988). A unified approach to controlling chaos via an lmi-based fuzzy control system design, *IEEE Transactions Circuits and Systems - I: Fundamental Theory and Applications* **45**(10): 1021–1040.
- Teixeira, M. C. M., Lordelo, A. D. S. & Assunção, E. (2000). On lmi based of spr systems and output variable structure controllers, *6th International Workshop on Variable Structure Systems (VSS'2000)*, Coolangatta Gold Coast, Australia, pp. 199–208.

- Teixeira, M. C. M., Pietrobon, H. C. & Assunção, E. (2000). Novos resultados sobre a estabilidade e controle de sistemas não-lineares utilizando modelos fuzzy e lmi, *Controle & Automação* **11**(1): 37–48.
- VanAntwerp, J. G. & Braatz, R. D. (2000). A tutorial on linear and bilinear matrix inequalities, *Journal of Process Control* **10**: 363–385.

Apêndice

O toolbox de LMI do MATLAB (P. Gahinet 1995) possui um editor gráfico para edição de LMIs, o *lmiedit*. A seguir mostramos um exemplo de utilização do editor gráfico de LMIs. Deseja-se editar as seguintes LMIs:

$$\begin{array}{rcl} A'P + PA & < & 0 \\ P & > & 0 \end{array}$$

com $P = P'$. Lembre-se de que A é uma matriz de elementos constantes e P é a variável do problema.

Para usar o recurso gráfico, digite *lmiedit* no espaço de trabalho do MATLAB. Aparecerá a tela de trabalho do **LMI Editor**. Digite o que está em negrito nos respectivos campos:

- *variable name*: **P**
- *type*: **S**
- *structure*: **[3 1]**
- *describe the LMIs as MATLAB*: **A'*P+P*A<0**
P>0

Selecione *view command*, perceba que foram listadas as declarações das variáveis e das LMIs, no formato de programação do MATLAB. Agora copie os comandos gerados pelo *lmiedit* no seu programa MATLAB.