Lista 2021 – Parte I

2021.1 Seja o sistema linear incerto

$$\delta(x) = A(\alpha)x,$$

sendo $A(\alpha)$ uma matriz politópica, isto é,

$$A(\alpha) = \sum_{i=1}^{N} \alpha_i A_i, \quad \alpha \in \Lambda_N$$

cujos vértices A_i, $i = 1, \ldots, N$ são conhecidos e Λ_N é o simplex unitário dado por

$$\Lambda_N \triangleq \left\{ \xi \in \mathbb{R}^N : \sum_{i=1}^{N} \xi_i = 1,\ \xi_i \geq 0,\ i = 1, \ldots, N \right\}$$

O operador $\delta(\cdot)$ representa a derivada ($\delta x(t) = d(x(t))/dt$) no caso de sistemas a tempo contínuo e o operador deslocamento unitário ($px(k) = x(k+1)$) no caso de sistemas a tempo discreto. Elabore uma rotina em matlab que desenhe o lugar das raízes da matriz $A(\alpha)$ testando diversos valores de α (malha fina). A rotina deve receber como entrada as matrizes A_i em formato célula ($A\{i\}$) e um *flag* indicando se é um sistema contínuo ou discreto. A malha fina deverá contemplar, no mínimo, os seguintes conjuntos de pontos

- 1000 pontos uniformemente distribuídos dentro do politopo. Para gerar esses pontos, use o código apresentado no apêndice.
- 100 pontos (igualmente espaçados) para cada segmento de reta entre quaisquer par de vértices (i, j) do politopo.
- 150 pontos uniformemente distribuídos dentro do subpolitopo formado por cada tripla (i, j, k) de vértices do politopo. Observação: não se aplica a sistemas com dois e três vértices.

Plote também os eixos real e imaginário e no caso de sistemas discretos, o círculo unitário. Como saída, a rotina deve fornecer o máximo valor da parte real (do módulo) dos autovalores, bem como o ponto α^* associado. Entregue o código fonte e dois exemplos gerados aleatoriamente de modo que o autovalor de maior parte real (maior módulo) não aconteça nos vértices ($\alpha_i = 1$, $\alpha_j = 0,\ i \neq j$).

2021.2 Programe as condições do Teorema 1 e Lemmas 1-4. Utilizando as condições programadas e o programa desenvolvido no exercício anterior, teste a estabilidade robusta dos sistemas lineares incertos contidos na base de dados

`http://wwww.dt.fee.unicamp.br/~ricfow/IA892/base2020.mat`

O seguinte código deve ser utilizado para realizar a leitura dos sistemas contidos na base

```
load('base2020.mat');
for i=1:size(BASE,2);
    A =  BASE{i};
    n = size(A{1},1);
    N = size(A,2);
    % aplique aqui o teste baseado em grid e as condições de estabilidade

end
```

Os resultados devem ser apresentados da seguinte maneira

sistema			T1			L1			L2			L3			L4		
n	N	λ_{max}	E	V	L	E	V	L	E	V	L	E	V	L	E	V	L
2	2	-0.2043	0	3	6	1	14	12	0	10	12	1	6	10	1	22	16
3	3	-0.0067	0	6	12	1	36	27	1	27	27	0	18	27	1	72	42
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

sendo que λ_{max} é a parte real do autovalor com maior parte real $\forall \alpha \in \Lambda_N$, computado pelo programa desenvolvido no primeiro exercício. As colunas n e N representam o número de estados e vértices, respectivamente. As colunas V e L indicam o número de variáveis escalares e linhas de LMIs, respectivamente. A coluna E informa se o sistema foi detectado como estável (1) ou não (0). Se existirem sistemas não certificados como estáveis por nenhuma das condições, é possível afirmar que os mesmos são instáveis? O que poderia ser feito para tentar provar a estabilidade desse(s) sistema(s)? Os códigos fontes das condições de estabilidade programadas devem ser apresentados.

2021.3 Seja a LMI dependente de parâmetros

$$A(\alpha(t))P(\alpha(t)) + P(\alpha(t))A(\alpha(t))' - \dot{P}(\alpha(t)) + B(\alpha(t))B(\alpha(t))' < 0$$

(a) Usando o Lema de Finsler, obtenha uma condição equivalente com variáveis de folga.

(b) Usando novamente o Lema de Finsler, obtenha uma LMI equivalente à condição obtida em (a) com mais variáveis de folga.

2021.4 Determine as normas $\mathcal{H}_2$ e $\mathcal{H}_\infty$ do sistema

$$\begin{cases} \dot{x} = -x + 3w \\ y = x \end{cases}$$

usando a definição. Compare os valores determinados com os obtidos por meio do comando `norm` do matlab.

2021.5 Seja o sistema

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bw(t) \\ y(t) &= Cx(t) + Dw(t) \end{aligned} \qquad (1)$$

com as seguintes matrizes

$$A = \begin{bmatrix} 0 & 1 \\ -16 & -1.2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 2 \end{bmatrix}, \quad D = 0$$

(a) Considerando $w(t) = \sin(\omega t)\exp(-0.1t)$, $t \in [0, 10]$, faça um programa que computa a relação entre a energia do sinal de saída $y(t)$ e a energia do sinal de entrada $w(t)$. O cômputo da energia de um sinal pode ser feito utilizando a rotina *trapz*, por exemplo

```
w=3;
T=0.01;
t=0:T:10;
sinal=sin(w*t).*exp(-0.1*t);
energia=sqrt(trapz(sinal.^2)*T);
```

(b) Compute a norma $\mathcal{H}_\infty$ do sistema usando LMIs (apresente o código) e depois teste o programa elaborado para os seguintes valores de $\omega \in \{3.8, 3.85, 3.9, 3.95, 4.0, 4.05, 4.10, 4.15, 4.20\}$.

(c) Desenhe o Diagrama de Bode e determine a norma $\mathcal{H}_\infty$ por inspeção, isto é, determinando o valor do pico (use a ferramenta de *zoom* para aumentar a precisão da inspeção). O valor obtido é próximo do computado no item (b)? Se não, o que poderia ser feito para diminuir a diferença?

(d) Substitua o sinal de entrada pelo impulso unitário e compute a norma $\mathcal{H}_2$ por meio da determinação da energia da saída.

(e) Compute a norma $\mathcal{H}_2$ do sistema usando LMIs (apresente o código) e compare o resultado com o obtido no item (d).

Referências

[1] A. Moeini, B. Abbasi, and H. Mahlooji. Conditional distribution inverse method in generating uniform random vectors over a simplex. *Communications in Statistics – Simulation and Computation*, 40(5):685–693, 2011.

Apêndice

Código que pode ser utilizado para encontrar pontos randômicos uniformemente distribuídos dentro do simplex unitário. Fonte: [1]

```
N=3 % número de vértices
x(1)=1−rand^(1/(N−1))
for k=2:N−1
    x(k)=(1−sum(x(1:k−1)))*(1−rand^(1/(N−k)));
end
x(N)=1−sum(x(1:N−1))
```